

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

«На правах рукопису»
УДК 004.45

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«__» _____ 20__ р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Програмне забезпечення інформаційно-
комунікаційних систем»
зі спеціальності 121 «Інженерія програмного забезпечення»
на тему: «Система управління підсвіткою рекламних оголошень»**

Виконав (-ла):
студент (-ка) VI курсу, групи ІТ-393мп
Кошовий Тимур Едуардович _____

Керівник:
Доцент кафедри АУТС, к.т.н., доцент
Катін Павло Юрійович _____

Рецензент:
доцент кафедри ОТ, к.т.н., доцент
Волокита Артем Миколайович _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Кошовому Тимуру Едуардовичу

1. Тема дисертації «Система управління підсвіткою рекламних оголошень», науковий керівник дисертації Катін Павло Юрійович, доцент кафедри АУТС, к.т.н., доцент, затверджені наказом по університету від «26» 10 2020 р. №3132-с

2. Термін подання студентом дисертації 21 грудня 2020

3. Об'єкт дослідження підсвітка рекламних оголошень, яка керується за допомогою мікроконтролера

4. Вихідні дані система управління підсвіткою рекламних оголошень

5. Перелік завдань, які потрібно розробити _____

1) Опис предметної області

2) Проведення огляду наявних аналогів та визначення технологій розробки

3) Проектування і розробка компонентів системи

4) Тестування розробленої системи

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: діаграма розгортання, діаграма компонентів, діаграма варіантів використання, діаграма послідовності, схема зон підсвітки, схема електрична принципова, схема структурна системи, схема функціональна

7. Орієнтовний перелік публікацій: наукова стаття на конференції Winter InfoCom Advanced Solutions 2020

8. Дата видачі завдання 29 квітня 2020

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Збір інформації	01.06.2020	
2.	Вивчення та аналіз завдання	12.08.2020	
3.	Підготовка публікацій	15.09.2020	
4.	Підготовка доповідей на конференції	20.09.2020	
5.	Підготовка дисертації	15.10.2020	
6.	Розробка системи	01.12.2020	
7.	Передзахист	02.12.2020	
8.	Захист	22.12.2020	

Студент

Тимур КОШОВИЙ

Науковий керівник

Павло КАТІН

РЕФЕРАТ

Магістерська дисертація: 101 с., 49 рис., 27 табл., 8 додатків, 21 джерело.

Актуальність. На сьогоднішній день рекламні оголошення є важливим інструментом комплексного маркетингу, вони утримують свою популярність вже друге десятиліття. Зазвичай на конструкції рекламного оголошення є зовнішня підсвітка, яка дозволяє ефективно впливати на покупців навіть в темний період доби. В результаті, запропоновану рекламу люди будуть бачити цілодобово та навіть у будь-яку погоду.

Зовнішня реклама – один з найбільш затребуваних інструментів маркетингу. Щоб рекламні щити ефективно працювали не тільки вдень, але і в темний час доби, необхідно подбати про якісне освітлення. Яскраво підсвічена реклама на нічних вулицях або магістралях виглядає ефектно, вона захоплює увагу споживача і сильніше впливає на нього

Метою роботи є підвищення якості реклами, покращення загального вигляду рекламних оголошень та збільшення кількості зацікавлених осіб.

Призначенням розробки є забезпечення гармонійного сприйняття кольорів як у денний, так і в нічний час та спрощення процесу залучення майбутніх клієнтів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- 1) проведення огляду програмних засобів розробки системи управління підсвіткою рекламних оголошень;
- 2) вивчення існуючих систем управління підсвіткою рекламних оголошень;
- 3) проектування і розробка системи;
- 4) реалізація системи;
- 5) тестування розробленої системи.

Об'єктом є підсвітка рекламних оголошень, яка керується за допомогою мікроконтролера.

Предметом є кросплатформна система керування підсвіткою рекламних оголошень.

Публікації. Результати роботи опубліковані у тезах X Міжнародної науково-практичної конференції Winter InfoCom Advanced Solutions 2020.

ПІДСВІТКА, РЕКЛАМНІ ОГОЛОШЕННЯ, ARDUINO, МІКРОКОНТРОЛЕР, RUST, ПОСЛІДОВНІ ПОРТИ, СВІТЛОДІОДИ

ABSTRACT

Master's Thesis: 101 pages, 49 fig., 27 tab., 8 app., 21 sources.

Topicality. Today, advertisements are an important tool of complex marketing, they have maintained their popularity for the second decade. Usually, the design of the advertisement has external lighting, which allows you to effectively influence customers, even in the dark. As a result, people will see the proposed advertisement around the clock and even in any weather.

Outdoor advertising is one of the most popular marketing tools. In order for billboards to work effectively not only during the day, but also in the dark, it is necessary to take care of lighting quality. Brightly lit advertising on the night streets or highways looks spectacular, it captures the attention of the consumers and has a stronger effect on them.

The aim of the work is to improve the quality of advertising, improve the overall appearance of advertisements and increase the number of interested people.

The purpose of the development is to ensure a harmonious perception of colors both during the day and at night and to simplify the process of attracting future customers.

To achieve this goal it is necessary to solve the following tasks:

- 1) review of software tools for the development of advertising management system;
- 2) study of existing control systems for illumination of advertisements;
- 3) design and development of the system;
- 4) implementation of the system;
- 5) testing of the developed system.

The object is the illumination of advertisements, which is controlled by a microcontroller.

The subject is a cross-platform control system for lighting.

Publications. The results are published in the abstracts of the X International Scientific and Practical Conference Winter InfoCom Advanced Solutions 2020.

LIGHTING, ADVERTISING, ARDUINO, MICROCONTROLLER, RUST, SERIAL PORTS, LEDS

ЗМІСТ

Вступ.....	10
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	12
1.1 Опис предметного середовища	12
1.2 Огляд наявних аналогів	14
1.3 Постановка задачі	22
1.3.1 Призначення розробки	22
1.3.2 Мета та задачі розробки	22
Висновки до розділу	22
2 ОПИС ТЕХНОЛОГІЙ	23
2.1 Послідовні порти	23
2.2 Світлодіодні стрічки	24
Висновки до розділу	26
3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	27
3.1 Засоби розробки	27
3.2 Вимоги до технічного забезпечення	37
3.2.1 Загальні вимоги	37
Висновки до розділу	37
4 РЕАЛІЗАЦІЯ СИСТЕМИ ТА РОЗРОБКА ПРОГРАМИ	39
4.1 Бібліотеки мови програмування Rust для взаємодії з послідовними портами	39
4.1.1 Бібліотека serial	39
4.1.2 Бібліотека serial-core	40
4.1.3 Бібліотека serial-windows	41
4.1.4 Бібліотека serial-unix	41
4.2 Бібліотека мови програмування Rust для захоплення зображення екрану	42

4.3	Опис структурної та функціональної моделі	43
4.3.1	Опис структурної моделі	43
4.3.2	Опис функціональної моделі	44
4.4	Реалізація драйвера мікроконтролера.....	44
4.5	Реалізація програмного забезпечення для операційної системи.....	50
4.5.1	Використання програми AmbiBox	50
4.5.2	Реалізація програмного забезпечення	56
4.6	Керівництво користувача	64
4.6.1	Налаштування мікроконтролера Arduino.....	64
4.6.2	Налаштування AmbiBox	70
4.6.3	Налаштування власного програмного забезпечення	73
4.6.4	Робота підсвітки	75
	Висновки до розділу	78
5	РОЗРОБКА СТАПТАП-ПРОЕКТУ	79
5.1	Опис ідеї проекту.....	79
5.1.1	Зміст ідеї	79
5.1.2	Аналіз техніко-економічних переваг ідеї.....	79
5.2	Технологічний аудит ідеї проекту	81
5.3	Аналіз ринкових можливостей запуску стартап-проекту.....	82
5.4	Розроблення ринкової стратегії проекту	90
5.5	Розроблення маркетингової програми стартап-проекту.....	94
	Висновки до розділу	98
	ВИСНОВКИ	99
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	100

ПЕРЕЛІК СКОРОЧЕНЬ

API	Application Programming Interface
COM	Communication port
DI	Digital input
DO	Digital output
FPS	Frames per second
IDE	Integrated Development Environment
OS	Operating system
OSD	On Screen Display
PWM	Pulse-width modulation
RGB	Red Green Blue

ВСТУП

Рекламні оголошення швидко увійшли в інструменти комплексного маркетингу і утримують свою популярність вже друге десятиліття. Рекламні оголошення дозволяють розміщувати зовнішню рекламу поблизу від місця продажу товару, що рекламується, таким чином збільшуючи кількість покупців. Зазвичай на цих конструкціях є зовнішня підсвітка, яка дозволяє ефективно впливати на покупців навіть в темний період доби. В результаті запропоновану рекламу будуть бачити люди у будь-яку погоду та навіть цілодобово [1].

Зовнішня реклама – один з найбільш затребуваних інструментів маркетингу. Щоб рекламні щити ефективно працювали не тільки вдень, але і в темний час доби, необхідно подбати про якісне освітлення. Яскрава підсвічена реклама на нічних вулицях або магістралях виглядає ефектно, вона захоплює увагу споживача і сильніше впливає на нього [2].

Рівномірність світла – одна з головних вимог до рекламного освітлення. Якщо на поверхні реклами присутні світлові плями, це негативно позначається на її сприйнятті. Домогтися ідеального рівномірного освітлення дозволяють проект освітлення: точні розрахунки потужності використовуваних приладів, їх кількості та розташування.

Щоб зовнішня реклама була яскравою і такою, що запам'ятовується, важливо забезпечити гармонійне сприйняття кольору як в денний, так і в нічний час. Для цього слід використовувати якісне обладнання з високим коефіцієнтом передачі кольору, що виключає спотворення кольору рекламного оголошення.

Особливе значення має оптимальна яскравість, так як тьмяне освітлення не приверне уваги, а надто яскраве підсвічування реклами буде сліпити і дратувати, більше того, на автомагістралях вона може бути небезпечною [2].

Рекламні оголошення здійснюють ефективний вплив на транспортні, а також пішохідні потоки.

Також серед основних переваг таких виробів – великий розмір, можливість установки біля автомобільних доріг з високим пасажиропотоком, а також

допустимість монтажу в безпосередній близькості від рекламованого закладу або об'єкта.

Підсвітка зовнішньої реклами має цілий ряд відчутних переваг:

- тривалий термін служби світлодіодів зводить до нуля експлуатаційні витрати;
- високий клас захисту від пилу і вологи визначає стійкість до зовнішніх впливів;
- допустимий діапазон зовнішніх температур (від -50 до +50 ° C) не накладає обмежень на регіон використання рекламних конструкцій;
- висока стійкість до вібрації усуває проблему великих вітрових навантажень;
- різноманітність колірних температур і високий індекс передачі кольору[3].

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Рекламні оголошення сучасного вигляду беруть початок в 20-30-х роках минулого століття, коли в США велася масова малоповерхова забудова, і біля кожного котеджного містечка стали розміщувати подібні плакати. З тих пір утекло чимало води, змінилися принципи рекламування продукції, з'явилися нові технології, вийшли на новий технічний рівень світильники, необхідні для підсвічування щитів в нічний час, а сама інформація стала носити більш агресивний або провокаційний характер.

Зараз жителям великих міст стали вкрай звичні великі рекламні щити. Вони впливають на потенційних покупців цілодобово - не тільки вдень, освітлювані променями сонця, але і по ночах, коли в хід йде спеціальне підсвічування. Використання освітлювальних приладів дозволяє підприємцям демонструвати свої товари або рекламувати послуги в будь-який сезон і при будь-якій погоді. Наукові дослідження в області маркетингу показують, що подібні щити помітно впливають на транспортні та пішохідні потоки, часто змушуючи людей згортати не туди, куди вони прямували спочатку.

Сьогоднішній споживач без перебільшення вкрай розпещений вибором товарів і послуг: різноманітні моделі, бренди, кольори, розміри, ціни та додатковий сервіс від виробника. Щоб залишатися конкурентоспроможними на цьому ринку, компанії прагнуть йти в ногу з часом і максимально яскраво заявляти про себе буквально на кожному кроці. Це може звучати дещо сумно, проте в наш час недостатньо лише добре працювати і випускати якісну продукцію - необхідно ще й залучити покупців, пояснивши їм сутність своїх нововведень і отримується споживачем зручність.

Незважаючи на те, що оренда рекламної площі – не дешево задоволення, найбільші витрати кожна організація несе за статтею витрат на електроенергію і пов'язані з освітлювальними приладами заходами з обслуговування. Наявні в таких місцях потужні світлодіодні світильники досить швидко забруднюються природним чином, а тому регулярно потребують професійної чистки захисного скла-розсіювача.

Крім того, природні явища і падаючі гілки періодично збивають прилади зі своїх місць, змушуючи монтажників замінювати їх на нові.

Звичайній людині здається, що володіння рекламним щитом – це досить простий бізнес з мінімальними інвестиціями і високою віддачею, що не вимагає додаткових зусиль з боку власника. Насправді ця сфера куди більш складна і конкурентна, а застосування прогресивних технологій багато в чому грає вирішальну роль. Воно або робить рекламну площу більш привабливою для потенційних рекламодавців, або знижує поточні прямі витрати на обслуговування.

Найтиповіший розмір білборда – 3 на 6 метрів, але існують компанії, які здатні виготовити щит з будь-яким співвідношенням сторін і роздрукувати за фрагментами підходить до нього постер. Стандартний метраж може бути досить ефективно освітлений всього двома-трьома прожекторами потужністю не більше 30 Вт кожен. Це один з найпоширеніших варіантів, що поєднують помірну собівартість і ефективність.

Одне з найважливіших вимог до подібного роду об'єктів полягає в рівномірності світла. Присутність світлових плям на рекламному плакаті буде вкрай негативно позначатися на тому, як спостерігач оцінить його зміст. Щоб домогтися оптимального розподілу світла, можуть використовуватися різні механізми: спеціальні розсіювачі, відбивачі, точки розстановки джерел світла та ін. Найбільш правильним буде підхід, що включає в себе попередній розрахунок освітлюваної площі, який враховує число приладів і точні місця їхнього розташування. Найчастіше рекламні плакати малюються таким чином, що основне їх зміст розташовується в центрі, а текст – зліва чи справа зверху. Такий стан речей зумовлює необхідність передбачити максимальну освітленість саме цих зон. Надалі облаштований рекламний щит стане практично універсальним для будь-якого типу розміщеної інформації [2].

1.2 Огляд наявних аналогів

У ході пошуку схожих за функціональністю систем управління підсвіткою рекламних оголошень виявлено наступні системи:

- розробки компанії «Азбука світла» [3];
- Arduino Pad, управління декоративною підсвіткою на Arduino з телефону [4];
- Helvar.com [5].

Опишемо особливості компанії «Азбука світла».

«Азбука кольору» створює комплексні світлотехнічні проекти, поставляє і виробляє світильники, впроваджує освітлення на архітектурних комерційних і спортивних об'єктах.

Компанія консультує архітекторів, дизайнерів, забудовників, які обслуговують компанії і приватних осіб щодо застосування світла для досягнення поставлених цілей. На рисунку 1.1 зображено сайт компанії.

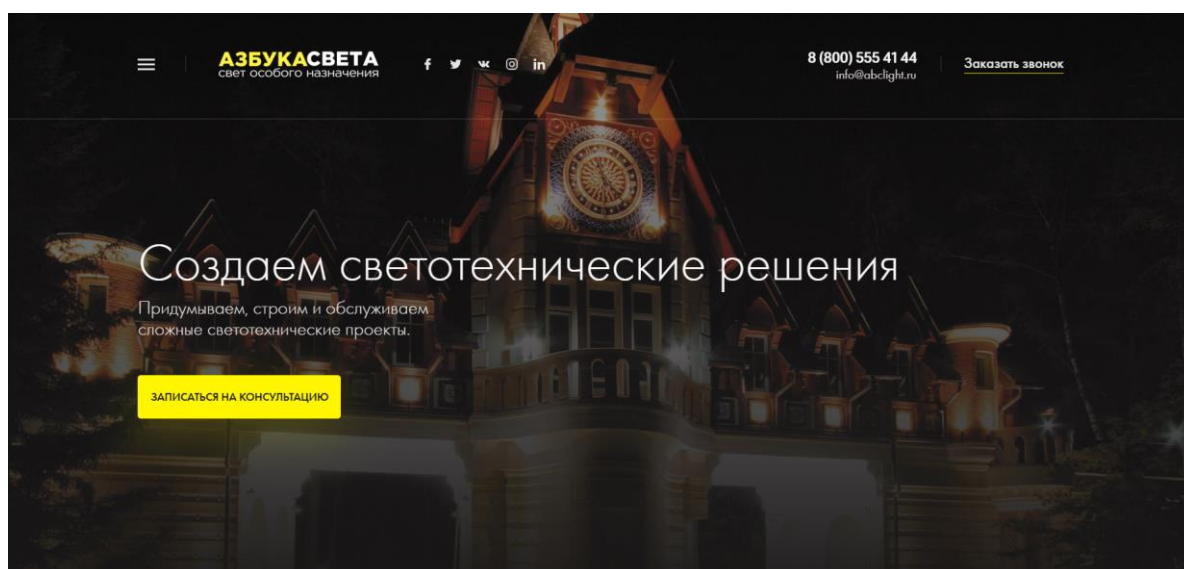


Рисунок 1.1 – Сайт компанії «Азбука кольору»

Історія компанії показана в таблиці 1.1.

Таблица 1.1 – Історія компанії «Азбука світла»

Рік	Подія
1993	Початок комерційної діяльності

Продовження таблиці 1.1

Рік	Подія
1997	Початок поставок світлотехніки з Європи
2000	Лідер продажів прожекторів Fael Luce (Італія) в Росії
2003	Початок виробництва металогалогенних прожекторів в Москві
2007	Початок виробництва перших світлодіодних прожекторів
2015	Створення відділу ремонтів та обслуговування світлодіодних систем
2019	Розробка нових моделей світильників, розширення партнерської мережі

Компанія довгі роки співпрацює з перевіреними підприємствами, які налагодили своє виробництво за кордоном, і готові пропонувати всі комплектуючі тільки європейської якості. Крім того, «Азбука світла» має в своєму арсеналі власне виробництво, забезпечуючи тим самим проектування та встановлення тільки оригінальних частин, металоконструкцій і освітлюваних об'єктів.

Компанія готова запропонувати послуги з проектування унікальних, дизайнерських рішень, які забезпечуються шляхом підбору унікальних освітлювальних приладів.

Головними напрямками в роботі компанії, є наступні:

- підбір та продаж світлового обладнання, яке забезпечує роботу всіх проектів по системі освітлення;
- проведення всіх необхідних робіт, які забезпечать повний цикл з проектування та облаштування освітлювальної системи;
- забезпечення освітлювальної системою об'єктів спортивного призначення, відкритих майданчиків, а також будівель і різних споруд;
- установка оригінальних конструкцій і подальше технічне обслуговування всіх систем для освітлення.

Компанія поділяє такі цінності:

- користь – насамперед: компанія дбає про клієнтів і допомагає досягати їх цілей

- потрібно поважати чужий час: краще виконати завдання заздалегідь, ніж підвести клієнта за термінами

- безпека понад усе: ми відповідаємо за результати своєї роботи і несемо відповідальність;

- фіксовані правила: компанія розповідає і показує результати своєї роботи.

У 2013 році «Азбука світла» випустила «Збірник типових світлотехнічних рішень». У збірнику приведені рекомендації щодо освітлення спортивних майданчиків, які пройшли рецензування на кафедрі світлотехніки Московського Енергетичного Інституту.

У 2019 ми випустили оновлений довідник, із застосуванням сучасного світлодіодного обладнання.

Розглянемо другий аналог – розробка драйверу управління декоративною підсвіткою на Arduino з телефону (Arduino Pad).

Задумка автора полягала в наступному. Розташовані вздовж стін світлодіоди знаходяться в отворах, просвердлених в ламінаті / паркеті, на невеликій відстані від стіни. При включенні - плавну зміну яскравості свічення окремих світлодіодів з одночасним зсувом уздовж стіни найбільш яскравого - на зразок посадочних вогнів на ЗПС.

Хотілося втілити в системі підсвічування:

- зміна швидкості і яскравості світіння, аж до повного відключення;
- режим незмінною яскравості.

Для управління було обрано телефон на Андроїд і Bluetooth, як найбільш простий спосіб, що не зав'язаний на систему команд певного пульта, не схильний до проблем передачі команд через перешкоди на зразок стін-меблів-котів.

Максимальна кількість світлодіодів - 6, за кількістю PWM виходів обраного Arduino.

Що потрібно доробити: можливість підключення модуля Bluetooth для прийому команд управління і відсилає даних про поточний стан на телефон для відображення.

Залежно від прийнятої команди контролер або змінює швидкість, або яскравість світіння світлодіодів. Так само можна запрограмувати будь-які довільні дії на прийняту команду - в прикладі я відключаю підсвічування і міняю швидкість.

У кожному циклі контролер шле по Bluetooth на парний пристрій рядок виду `### 0-50-100-150-200-250 ***`, де цифри - значення яскравості шести світлодіодів, з максимумом в 255.

На рисунку 1.2 приведено ілюстрацію підключення компонентів системи.

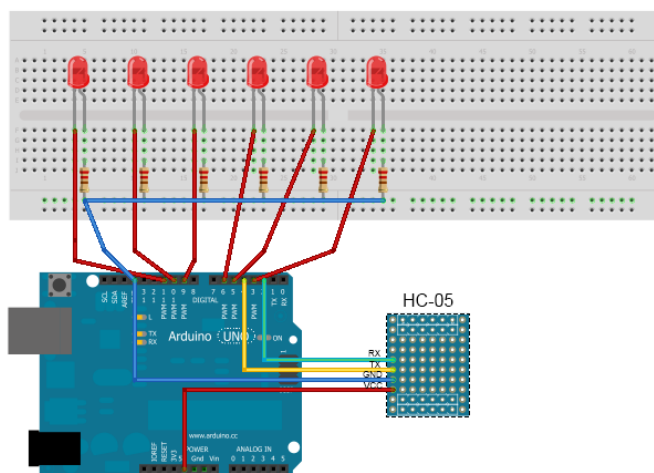


Рисунок 1.2 – Ілюстрацію підключення компонентів системи

З чим зіткнувся автор: неможливість прийому багатосимвольних команд від телефону. Наприклад, посылаючи команду `s150` (за задумом - встановити швидкість 150 одиниць при максимумі в 255), автор отримував на Ардуіно замість переданого рядка, не валідні дані виду `s ###`, де замість `###` йшли будь-які ASCII-символи. Можливо, це пов'язано з використанням класу `SoftwareSerial` для підключення Bluetooth. Пошуки в інтернеті показали, що авирп не єдиний хто зіткнувся з цією проблемою, але немає жодного рішення, окрім як зміна режимів модуля HC-05. Тому було вирішено використовувати тільки односимвольні команди з певних діапазонів, тим більше для поставленого завдання цього цілком вистачало. Наприклад - для швидкості в 10 одиниць і команди від G до P, для яскравості - діапазон від Q до Z відповідно.

Для телефону було написано додаток, з можливістю завдання переданих команд без переписування і компіляції коду - для універсальності. На рисунку 1.3 приведено скріншоти реалізованого додатку.

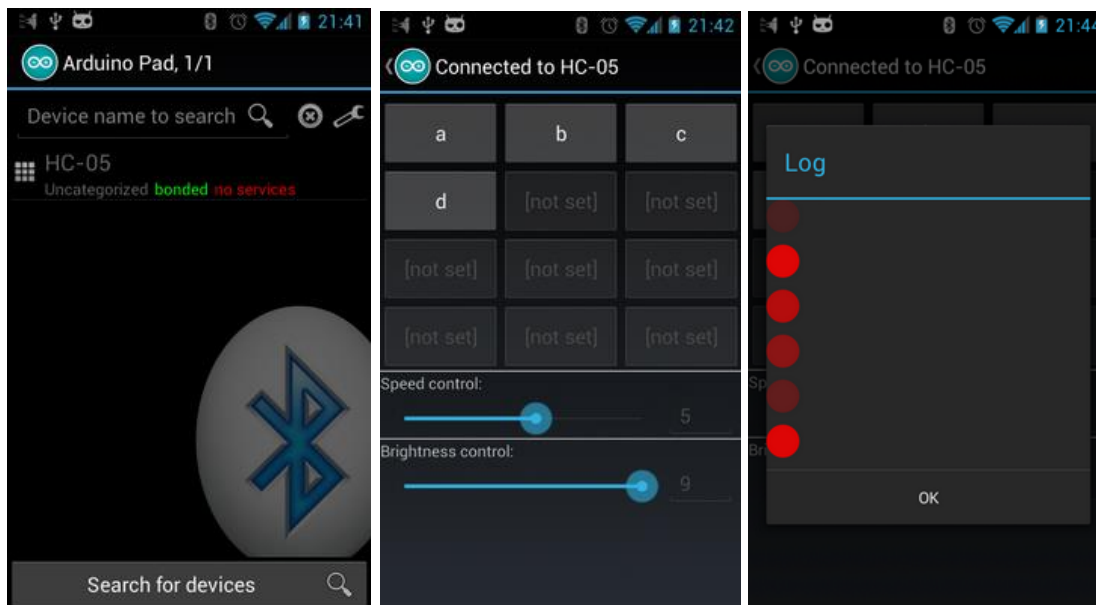


Рисунок 1.3 – Скріншоти реалізованого додатку

Власне остаточна пайка схеми і мідна електропроводка проводів-світлодіодів по кімнаті

Підключення модулів обсягу-присутності для включення / відключення системи в залежності від присутності людини в кімнаті

Розглянемо систему інтелектуальних світлових рішень Helvar.com.

На сьогодні Helvar.com – одна з значуща компаній Європи у своїй галузі, яка змогла унікальним чином поєднати досвід з розробки та керування освітленням завдяки своєму прагненню виробляти інноваційні продукти та рішення.

Її штаб-квартира знаходиться у Фінляндії, а також розробляються компоненти світильника компанії та власні виробничі операції. Центр компетентності системи освітлення знаходиться в Лондоні.

Міжнародність та експорт є основою Гельвару з самого початку, і 97% продажів сьогодні є за межами Фінляндії.

Незважаючи на вражаюче зростання та технологічний розвиток, компанія все ще має свій оригінальний «дух Гельвара» - сильну місцеву гордість, близькість до клієнтів, працьовитість та першокласне промислове ноу-хау.

На рисунку 1.4 зображено сайт компанії Helvar.

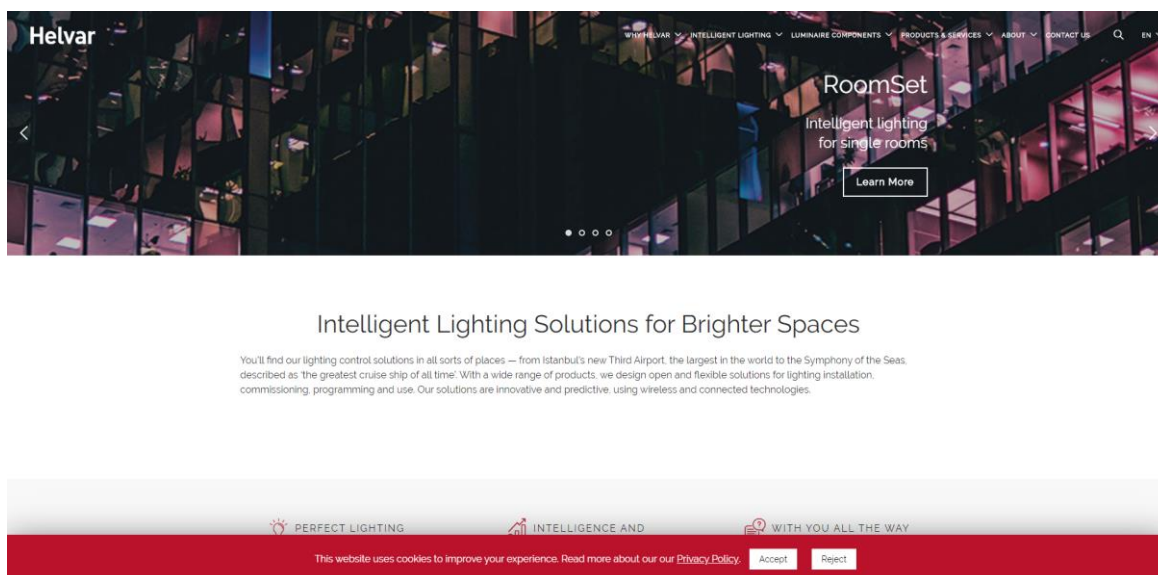


Рисунок 1.4 – Сайт компанії Helvar

Компанія пропонує ідею штучного інтелекту для освітлення.

Штучний інтелект використовується в різних сферах, широко застосовуючи світлотехнічну промисловість на різних етапах життєвого циклу техніки освітлення – від проектування до встановлення та введення в експлуатацію. Наприклад, нейронні системи автоматично освітлюють, налаштовуються, та вимикаються без взаємодії людей. Такою є економія часу на введення в експлуатацію нової освітлювальної техніки. Система освітлення на основі штучного інтелекту контролює і вимірює параметри внутрішніх срібла в приміщеннях. На підставі отриманих даних вона може оптимізувати і налаштовувати параметри освітлення у відповідності з потребами користувачів для забезпечення сприятливих умов.

Така система виходить не тільки з орендарами, а і з іншими сторонами, такими, як власники будівель та керівники об'єктів. Системи освячують з управлінням на базі даних, постійно генерують дані, які збираються та зберігаються на централізованому сервері. Алгоритми та штучний інтелект можуть бути

використані для реалізації у вихідних компонентах, наприклад, датчики, для прийняття децентралізованих рішень у режимі реального часу або на серверах для прийняття централізованих рішень. Крім того, ці дані можуть бути використані для інших систем управління (BMS), як опалення, вентиляція та кондиціонування повітря (HVAC), або для розумним будинком.

Проте, застосовуючи цю технологію, потрібно з обережністю. Камери з технологіями та штучним інтелектом можуть точно зареєструвати нагляд і переміщення об'єктів у комерційному або просторовому просторі, а в організаціях візуального спостереження необхідно суворо дотримуватися законів про недоторканність приватного життя. Отже, фактично існує безліч різноманітних прикладів штучного інтелекту у світлотехнічній промисловості, і такі рішення приносять практичну вишукану користь, підвищують комфорт, виробництво і навіть дозволяють рентабельність за умови, що відповідає законам про недоторканність людини.

Проста нейронна система здатна істотно економити енергію. У світильників на основі рішень ActiveAhead використовує датчики для забезпечення інтелектуального освітлення та збору даних та оптимізації освітлення у цілому.

Одне просте рішення, адаптується до зміни розміщення використання простору, забезпечує комфорт і енергоефективність.

Для офісів, коридорів, автостоянок та проектів реконструкції економія енергії досягає 20–80%.

В таблиці 1.2 наведено короткий опис характерних особливостей існуючих додатків.

Таблиця 1.2 – Характеристика існуючих підходів

Назва	Опис
Розробки компанії «Азбука світла»	«Азбука кольору» створює комплексні світлотехнічні проекти, поставляє і виробляє світильники, впроваджує освітлення на архітектурних комерційних і спортивних об'єктах.

Продовження таблиці 1.2

Назва	Опис
	Компанія консулює архітекторів, дизайнерів, забудовників, які обслуговують компанії і приватних осіб щодо застосування світла для досягнення поставлених цілей. Ціль компанії: постійно вдосконалювати свої технології, розширювати свою діяльність і асортимент продукції.
Управління декоративною підсвіткою на Arduino з телефону	Розташовані вздовж стін світлодіоди знаходяться в отворах, просвердлених в ламінаті або паркеті, на невеликій відстані від стіни. При включенні - плавну зміну
Arduino Pad	яскравості свічення окремих світлодіодів з одночасним зсувом уздовж стіни найбільш яскравого - на зразок посадочних вогнів на ЗПС. Ціль розробки: полегшити управління світлом за допомогою використання мобільного пристрою.
Helvar.com	Компанія пропонує ідею штучного інтелекту для освітлення. Система освітлення на основі штучного інтелекту контролює і вимірює параметри рівня освітлення в приміщеннях. На підставі отриманих даних вона може оптимізувати і налаштовувати параметри освітлення у відповідності з потребами користувачів для забезпечення сприятливих умов. Ціль компанії: надавати клієнтам найвищу якість послуг та підтримки.

Як бачимо, усі розглянуті системи мають схожі функції та ідею, але кожен має свої недоліки та переваги, які впливають на загальне враження від системи.

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначенням розробки є забезпечення гармонійного сприйняття кольорів як у денний, так і в нічний час та спрощення процесу залучення майбутніх клієнтів.

1.3.2 Мета та задачі розробки

Метою роботи є підвищення якості реклами, покращення загального вигляду рекламних оголошень та збільшення кількості зацікавлених осіб.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- 1) проведення огляду програмних засобів розробки системи управління підсвіткою рекламних оголошень;
- 2) вивчення існуючих систем управління підсвіткою рекламних оголошень;
- 3) проектування і розробка системи;
- 4) реалізація системи;
- 5) тестування розробленої системи.

Висновки до розділу

У ході написання даного розділу були наведені загальні відомості про предметне середовище: актуальність проблеми, її необхідність. Було визначено роль освітлення та підсвіток рекламних оголошень. Також визначено, що рівномірність світла – одна з головних вимог до рекламного освітлення.

Розглянуто аналоги системи: розробки компанії «Азбука світла», система управління декоративною підсвіткою на Arduino з телефону та розробки компанії Helvar.com. Проведено аналіз аналогів інших компаній та розробників, виявлено відмінності між ідеями систем управління підсвітками, тим самим виявлено, що кожен з аналогів має свої особливості. Було визначено постановку задачі – призначення, мету. Для досягнення мети були визначені задачі для вирішення.

2 ОПИС ТЕХНОЛОГІЙ

2.1 Послідовні порти

Послідовний порт або COM-порт (від англ. COMmunication port) – двонаправлений послідовний інтерфейс, призначений для обміну байтовою інформацією.

Послідовним його називають тому, що інформація через нього передається у вигляді послідовності електричних імпульсів, біт за бітом (на відміну від паралельного порту). Найбільш часто для послідовного порту персональних комп'ютерів використовується стандарт RS-232C.

Раніше послідовний порт використовувався для підключення терміналу, пізніше для модему або миші. Зараз він використовується для з'єднання з джерелами безперебійного живлення, для зв'язку з апаратними засобами розробки вбудованих обчислювальних систем.

Хоча деякі інші інтерфейси комп'ютера – такі як Ethernet, FireWire і USB – також використовують послідовний спосіб обміну, назва "послідовний порт" закріпилося за портом, що має стандарт RS-232C, і призначеним для обміну інформацією з модемом. На рисунку 2.1 представлено послідовний порт стандарту RS-232.



Рисунок 2.1 – Послідовний порт стандарту RS-232

На рисунку 2.2 представлено інтерфейс обміну даними послідовного порту стандарту RS-232.

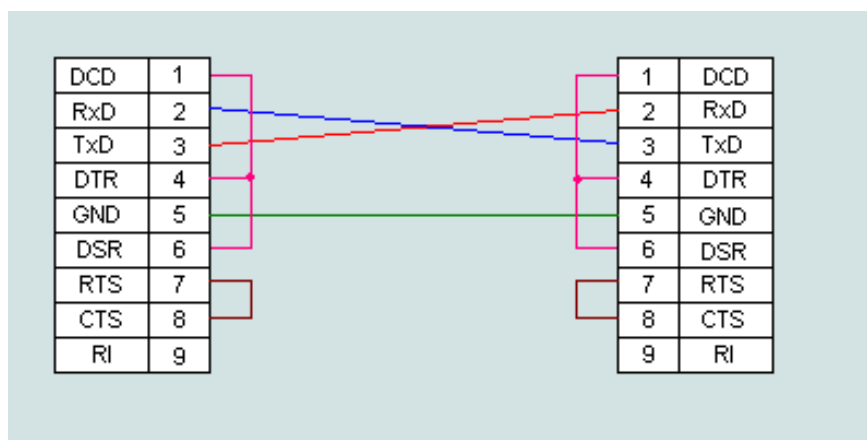


Рисунок 2.2 – Інтерфейс обміну даними послідовного порту стандарту RS-232

В даний час в персональних комп'ютерах все ще зустрічається даний вид інтерфейсу, незважаючи на значне витіснення іншими портами: PS / 2 (підключення миші і клавіатури), USB універсальна послідовна шина з живленням [6].

2.2 Світлодіодні стрічки

Звичайна світлодіодна стрічка представляє собою стрічку зі світлодіодами і резисторами, живлення відбувається за допомогою двох дротів: плюса та мінуса. Напруга буває різною: 5 і 12 вольт постійного струму, або 220 вольт змінного. Для стрічок які живляться напругою 5 або 12 вольт стрічок необхідно використовувати блоки живлення. Світить така стрічка одним кольором, який залежить від світлодіодів.

RGB світлодіодна стрічка. На цій стрічці стоять RGB світлодіоди. Такий світлодіод має вже 4 виходи, один загальний плюс, і три мінуса на кожен колір, тобто всередині одного світлодіода знаходиться три світлодіода різних кольорів. Подаючи живлення на загальний плюс і будь-який з кольорів, ми вмикаємо цей колір. Подамо на всі три – отримаємо білий, зелений і червоний дадуть жовтий, і так далі. Для таких стрічок існують контролери з пультами, типовий контролер є три транзистора на

кожен колір і мікроконтролер, який керує транзисторами, таким чином даючи можливість включити будь-який колір. І, як ви вже зрозуміли, так, керувати такою стрічкою з Ардуїно дуже просто.

Адресна світлодіодна стрічка являє собою стрічку з адресних діодів, один такий світлодіод складається з RGB світлодіода та контролера. В середині кожного світлодіода знаходиться контролер з трьома транзисторними виходами. Завдяки цьому у нас є можливість керувати кольором (тобто яскравістю кожного з кольорів) будь-якого світлодіода в стрічці і створювати приголомшливі ефекти. Адресна стрічка може мати 3 або 4 контакти для підключення, два з них завжди живлення (5V і GND наприклад), і інші (один або два) – логічні, для управління.

Команди в стрічці передаються послідовно від діода до діода. Стрічка має початок і кінець, напрямок руху команд на деяких моделях зазначено стрілочками. Для прикладу розглянемо стрічку ws2812b (рисунок 2.3), вона має три контакти. Два на живлення та один логічний. На початку стрічки логічний контакт називається DI (digital input), а в кінці – DO (digital output). Стрічка бере команди в контакт DI. Контакт DO потрібен для підключення додаткових шматків стрічки або з'єднання матриць. Максимальний струм одного адресного світлодіода дорівнює 60 міліампер. Робочі температури лежать в межах від -25 до +80 градусів. Напруга живлення становить $5\text{ В} \pm 0,5$.

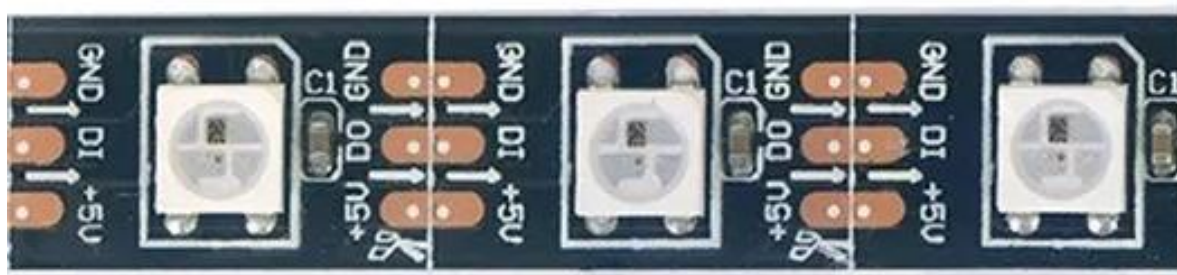


Рисунок 2.3 – Світлодіодна стрічка ws2812b

Основні переваги стрічки на основі ws2812b:

- компактні розміри;

- легкість в управлінні;
- управління здійснюється лише по одній лінії + дроти живлення;
- кількість включених послідовно світлодіодів не обмежена;
- невисока вартість – покупка окремо трьох світлодіодів і драйвера до них

вийде значно дорожче.

Основні недоліки:

- при виході з ладу хоча б одного світлодіода, всі світлодіоди, які розташовані після нього перестають працювати [7].

Висновки до розділу

У даному розділі були розглянуті теоретичні засади послідовних портів та світлодіодних стрічок.

Визначено, що послідовний порт – це двонаправлений послідовний інтерфейс, призначений для обміну байтовою інформацією. Представлено інтерфейс обміну даними послідовного порту стандарту RS-232.

Наведено характеристики адресної світлодіодної стрічки. Вказано, що адресна світлодіодна стрічка являє собою стрічку з адресних діодів, один такий світлодіод складається з RGB світлодіода та контролера.

Описано принцип роботи на прикладі стрічки ws2812b, а також її основні переваги та недоліки.

3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

При розробці драйвера мікроконтролера для плати Arduino Nano було використано такі засоби для програмування на Arduino як Arduino IDE.

Мова програмування Arduino – досить проста в освоєнні, так як основною цільовою аудиторією його застосування є любителі. Однак вважається однією з найкращих мов для програмування мікроконтролерів [8]. Фактично, мова програмування пристроїв Arduino заснована на C/C++ і скомпонована з бібліотекою AVR Libc і дозволяє використовувати будь-які її функції.

Мови C та C++ навіть на сьогоднішній день, незважаючи на все різноманіття сучасних надійних інструментів, залишаються незамінними. C і C++ гнучкі, дуже швидкі та ефективні. До того ж вони ближчі до машинного коду, ніж будь-яка інша високорівнева мова програмування. Саме тому в деяких випадках ці мови просто незамінні. C – друга мова за популярністю, яка незначно поступається Java. За передбаченнями, в найближчому майбутньому мови C та C++ не зійдуть з технологічної арени. Особливо через зростання кількості пристроїв, що підтримують технологію Інтернету речей, а ці пристрої міцно зав'язані на управлінні машинним кодом і пам'яттю.

Мови C/C++ цілком підходять для:

- розробки операційних систем;
- програмування вбудованих систем;
- розробки ігор;
- розробки баз даних;
- розробки десктопних і кроссплатформених додатків.

Також можна відзначити, що в проектах, що мають справу зі складними типами даних, розробка програмного забезпечення з використанням C++ запобігає перевантаженню даними та справляється зі схожими проблемами [9].

AVR Libc – версія стандартної бібліотеки C, враховує архітектурні особливості платформи AVR, зокрема, гарвардську архітектуру.

AVR – сімейство восьмибітних мікроконтролерів, що раніше випускалися фірмою Atmel, потім Microchip. Рік розробки – 1996.

Мікроконтролери AVR мають гарвардську архітектуру (програма і дані знаходяться в різних адресних просторах) і систему команд, близьку до ідеології RISC (англ. reduced instruction set computer – комп'ютер з набором коротких (простих, швидких) команд).

RISC – архітектура процесора, в якій швидкодія збільшується за рахунок спрощення інструкцій, щоб їх декодування було простішим, а час виконання – меншим.

Процесор AVR має 32 8-бітових регістра загального призначення, об'єднаних в регістровий файл. На відміну від «ідеального» RISC, регістри не абсолютно ортогональні [10].

Так як програмування мікроконтролерів Arduino здійснюється на мові програмування C/C++, проте використовується спрощена версія цієї мови програмування. Так само для спрощення розробки прошивок існує безліч функцій, класів, методів і бібліотек. Завдяки цьому працювати з цими мікроконтролерами дуже зручно і легко. Опис більшості необхідних функцій і операторів знаходиться у відкритому доступі. Цього вистачить для написання прошивок під найрізноманітніших пристрої на базі Arduino. Для всіх функцій є невеликі приклади, що показують як їх можна використовувати [11].

Структура програми Arduino досить проста і в мінімальному варіанті складається з двох частин. Перша частина, функція `setup()` яка виконується один раз при запуску програми – рисунок 3.1, та функція `loop()` яка виконується циклічно – рисунок 3.2.

```
void setup()  
{  
  
}
```

Рисунок 3.1 – Функція `setup()`

```
void loop() {  
  
}
```

Рисунок 3.2 – Функція loop()

Функція `setup()` виконується один раз, при включенні живлення або скиданні контролера. Зазвичай в ній відбуваються початкові установки змінних, регістрів. Функція має бути присутня в програмі, навіть якщо в ній нічого немає.

Після завершення `setup()` управління переходить до функції `loop()`. Вона в нескінченному циклі виконує команди, записані в її тілі (між фігурними дужками). Власне ці команди і роблять всі алгоритмічні дії контролера.

Для початку роботи з Arduino необхідно встановити середовище програмування Arduino IDE.

Arduino IDE – інтегроване середовище розробки для Windows, MacOS і Linux, розроблена на C і C ++, призначена для створення і завантаження програм на Arduino-сумісні плати, а також на плати інших виробників.

Вихідний код для середовища випущений під загальнодоступною ліцензією GNU версії 2. Підтримує мови C і C ++ з використанням спеціальних правил структурування коду. Arduino IDE надає бібліотеку програмного забезпечення з проекту Wiring , яка надає безліч загальних процедур введення і виведення. Для написаного користувачем коду потрібні тільки дві базові функції для запуску ескізу і основного циклу програми, які скомпільовані і пов'язані з функцією `main()` в виконувану циклічну програму з ланцюжком інструментів GNU, також включеною в дистрибутив IDE. Вона використовує програму `avrdude` для перетворення коду, в текстовий файл в шістнадцятковій системі кодування, який завантажується в плату Arduino програмою-завантажувачем у вбудованому програмному забезпеченні плати.

З ростом популярності Arduino інші постачальники в якості програмної платформи почали впроваджувати призначені для користувача компілятори і інструменти з відкритим вихідним кодом (ядра), які можуть створювати і

завантажувати ескізи в інші мікроконтролери, що не підтримуються офіційною лінійкою мікроконтролерів Arduino [12].

Крім офіційної Arduino IDE, існують програми сторонніх розробників, які пропонують свої продукти для роботи з мікроконтролерами на базі Arduino.

Аналогічний набір функцій нам може надати програма, яка називається Processing. Вона дуже схожа з Arduino IDE, так як обидві зроблені на одному двигуні. Processing має великий набір функцій, який мало поступається оригінальній програмі. За допомогою бібліотеки Serial користувач може створити зв'язок між передачею даних, які передають один одному плата і Processing. При цьому ми можемо змусити плату виконувати програми прямо з нашого персонального комп'ютера.

Існує ще одна цікава версія вихідної програми. Називається вона B4R, і головною її відмінністю є використання в якості основи не мови C чи C++, а іншої мову програмування – Basic. Даний програмний продукт є безкоштовним.

Є і платні варіанти Arduino IDE. Одним з таких є програма PROGROMINO. Головною її перевагою вважається можливість автодоповнення коду. При складанні програми більше не потрібно шукати інформацію в довідниках. Програма сама запропонує можливі варіанти використання тієї чи іншої процедури. В її набір входить ще безліч цікавих функцій, відсутніх в оригінальній програмі і здатних полегшити роботу з платами.

Конкуренти Arduino

Даний ринок з виробництва мікроконтролерів для створення різних електронних схем і робототехніки має багато прихильників по всій земній кулі. Дана ситуація сприяє появі на ринку не тільки конкурентів, які пропонують схожі продукти. Крім них випускається значна кількість підробок різної якості. Одні дуже важко відрізнити від оригіналів, адже вони мають ідентичну якість, інші мають дуже погані характеристиками і можуть зовсім не працювати з оригінальними продуктами.

Існують навіть плати Arduino, які підтримують роботу мікропроцесорів з інтерпретаторами JavaScript. Адже вони більш прості, і дозволяють домогтися результатів з підвищеною швидкістю. Однак дані плати є більш дорогими по відношенню до Arduino, що є істотним мінусом [8].

Драйвер було розроблено для плати Arduino Nano. Arduino Nano – це повнофункціональний мініатюрний пристрій на базі мікроконтролера ATmega328 (Arduino Nano 3.0) або ATmega168 (Arduino Nano 2.x), адаптований для використання з макетними платами. За функціональністю пристрій схожий на Arduino Duemilanove, і відрізняється від нього розмірами, відсутністю роз'єму живлення, а також іншим типом (Mini-B) USB-кабелю. Arduino Nano розроблено і випускається фірмою Gravitech.

На рисунку 3.2 представлено елементи плати [13].

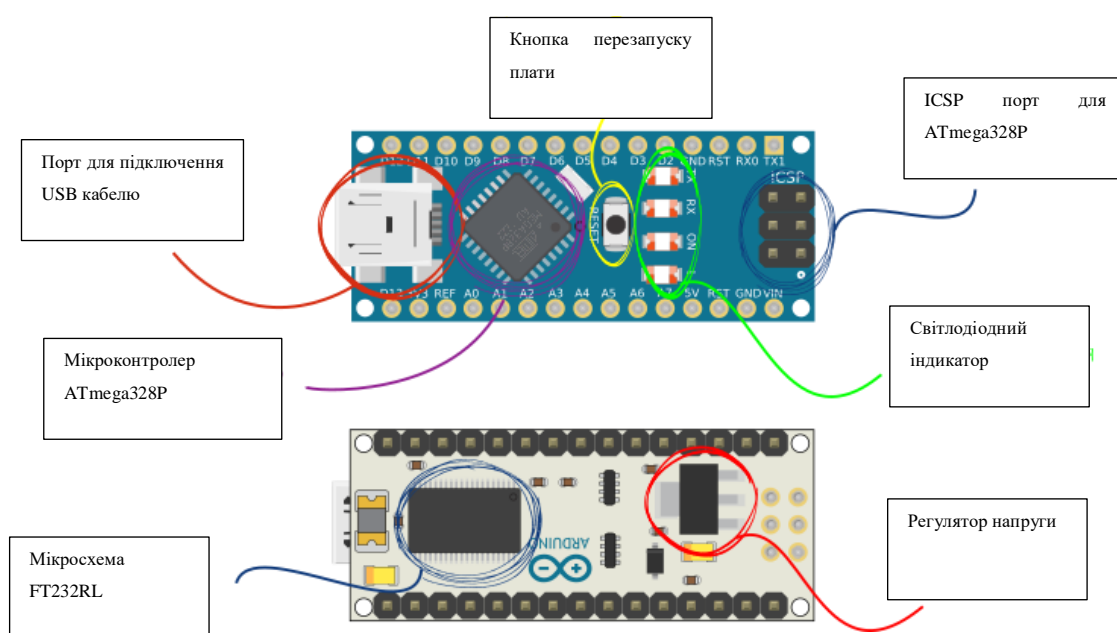


Рисунок 3.2 – Елементи плати Arduino Nano

Мікроконтролер ATmega328P

Серцем платформи Arduino Nano є 8-бітний мікроконтролер сімейства AVR – ATmega328P з тактовою частотою 16 МГц. Контролер надає 32 КБ Flash-пам'яті для зберігання прошивки, 2 КБ оперативної пам'яті SRAM і 1 КБ незалежній пам'яті EEPROM для зберігання даних.

Мікросхема FT232R

Мікросхема FTDI FT232R забезпечує зв'язок мікроконтролера ATmega328P з USB-портом комп'ютера. При підключенні до ПК Arduino Nano визначається як віртуальний COM-порт.

У таблиці 3.1 показано світлодіодну індикацію плати Arduino Nano.

Таблиця 3.1 – Світлодіодна індикацію плати Arduino Nano

Ім'я світлодіода	Призначення
RX та TX	Блимають при обміні даними між Arduino Nano і ПК.
L	Користувачський світлодіод, підключений до 13-го піну мікроконтролера. При високому рівні світлодіод включається, при низькому - вимикається.
ON	Наявність живлення на Arduino Nano.

Роз'єм Mini-USB

Роз'єм Mini-USB призначений для прошивки платформи Arduino Nano за допомогою комп'ютера.

Регулятор напруги 5 В

Лінійний понижуючий регулятор напруги LM1117MPX-5.0 з виходом 5 вольт забезпечує живлення мікроконтролера ATmega328P і іншої логіки платформи. Максимальний вихідний струм складає 800 мА.

ICSP-роз'єм для ATmega328

ICSP-роз'єм призначений для завантаження прошивки в мікроконтролер ATmega328 через програматор.

Через контакти ICSP Arduino Nano спілкується з платами розширення по інтерфейсу SPI (Serial Peripheral Interface, SPI bus — послідовний периферійний інтерфейс).

На рисунку 3.3 показана ілюстрація пінів платформи Arduino Nano – схема розташування виводів або розводка виводів – опис кожного контакту електричного з'єднання (наприклад, роз'єму) в електронній апаратурі [13].

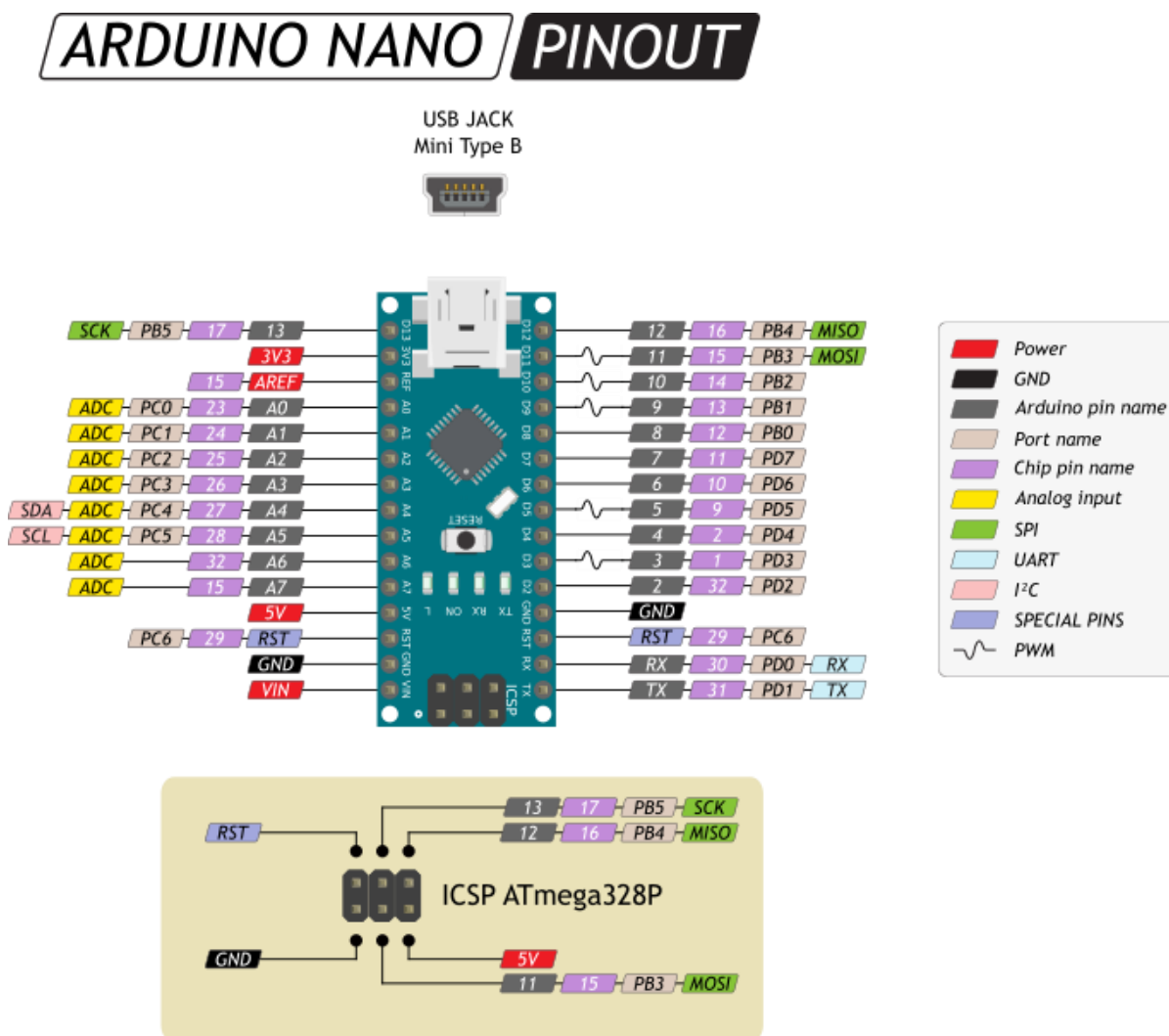


Рисунок 3.3 – Ілюстрація пінів платформи Arduino Nano

VIN: Вхідний пін для підключення зовнішнього джерела живлення з напругою в діапазоні від 7 до 12 вольт.

5V: Вихідний пін від регулятора напруги на платі з виходом 5 вольт і максимальних струмом 800 мА. Живити пристрій через висновок 5V не рекомендується - ви ризикуєте спалити плату.

3.3V: Вихідний пін від стабілізатора мікросхеми FT232R з виходом 3,3 вольт і максимальних струмом 50 мА. Живити пристрій через висновок 3V3 не рекомендується - ви ризикуєте спалити плату.

GND: Вивід землі.

AREF: Пін для підключення зовнішнього опорного напруги АЦП щодо якого відбуваються аналогові вимірювання при використанні функції `analogReference ()` з параметром «EXTERNAL».

Порти вводу/виводу

Цифрові входи/виходи: Піни 0-13

Логічний рівень одиниці – 5 В, нуля - 0 В. Максимальний струм виходу - 40 мА. До контактів підключені підтягує резистори, які за замовчуванням вимкнені, але можуть бути включені програмно.

ШІМ: Піни 3,5,6,9,10 і 11

Дозволяє виводити аналогові значення в вигляді ШІМ-сигналу. Розрядність ШІМ не змінюється і встановлена в 8 біт.

АЦП: Піни А0-А7

Дозволяє представити аналогову напругу в цифровому вигляді. Розрядність АЦП не змінюється і встановлена в 10 біт. Діапазон входної напруги від 0 до 5 В. При подачі більшої напруги - ви вб'єте мікроконтролер.

TWI / I²C: Піни А4 (SDA) і А5 (SCL)

Для спілкування з периферією по інтерфейсу I²C. Для роботи використовуйте бібліотеку Wire.

SPI: Піни 11 (MOSI), 12 (MISO), 13 (SCK) і 10 (SS)

Для спілкування з периферією по інтерфейсу SPI. Для роботи - використовуйте бібліотеку SPI.

UART: Піни 0 (RX) і 1 (TX)

Використовується для комунікації плати Arduino з комп'ютером або іншими пристроями по послідовному інтерфейсу. Висновки 0 (RX) і 1 (TX) з'єднані з відповідними USB-UART перетворювача FT232R. Для роботи з послідовним інтерфейсом - використовуйте методи бібліотеки Serial.

Принципова схема представлена на рисунку 3.4.

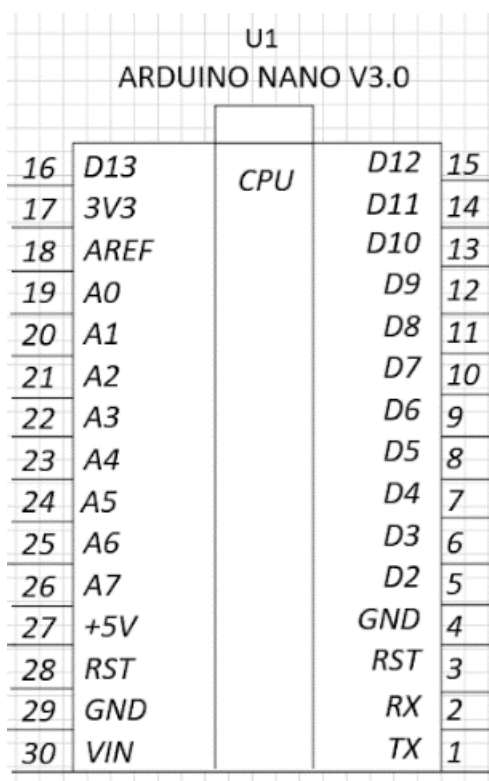


Рисунок 3.4 – Принципова схема платформи платформи Arduino Nano

Характеристики платформи Arduino Nano:

- мікроконтролер: ATmega328P;
- ядро: 8-бітний AVR;
- тактова частота: 16 МГц;
- flash-пам'ять: 32 КБ (2 КБ займає завантажувач);
- SRAM-пам'ять: 2 КБ;
- EEPROM-пам'яті: 1 КБ;
- портів введення-виведення всього: 20;
- портів з АЦП: 8;
- розрядність АЦП: 10 біт;
- портів з ШІМ: 6;
- розрядність ШІМ: 8 біт;
- апаратних інтерфейсів SPI: 1;
- апаратних інтерфейсів I²C / TWI: 1;
- апаратних інтерфейсів UART / Serial: 1;
- номінальна робоча напруга: 5 В;

- максимальний вихідний струм піна 5V: 800 мА;
- максимальний вихідний струм піна 3V3: 50 мА;
- максимальний струм з піна або на пін: 40 мА;
- допустима вхідна напруга від зовнішнього джерела: 7-12 В;
- габарити: 18 × 45 мм [13].

Для розробки програмного забезпечення для налаштування і персоналізації фонові підсвітки використано мову програмування Rust.

Rust – це універсальна мова програмування, що розробляється компанією Mozilla, три основних принципи якої: швидкість, безпека і ергономіка. Самі творці нескромно вважають її однією з найбільш ймовірних спадкоємців C/C++.

Rust досить складна для навчання. По-перше, це обумовлено молодістю мови і, як наслідок, малою кількістю літератури. По-друге, вивчити її людині далекій від програмування, можливо буде навіть простіше, ніж знайомому з іншими мовами. Так, наприклад, готового ІТ-фахівця буде сильно дратувати необхідність прописувати найменші операції, а відсутність як такого наслідування в мові і просто поставить в безвихідь.

Принципи роботи з пам'яттю Rust відчутно відрізняються як від мов з повним доступом до пам'яті, так і від мов з повним контролем за пам'яттю з боку збирача сміття. Модель пам'яті Rust побудована таким чином, що, з одного боку, надає розробнику можливість контролювати, де розміщувати дані, вводячи поділ за типами покажчиків і забезпечуючи контроль за їх використанням на етапі компіляції. З іншого боку, механізм підрахунку посилань Rust намагається видавати помилки компіляції в тих випадках, в яких використання інших мов призводить до помилок часу виконання або аварійного завершення програм.

Мова дозволяє оголошувати функції і блоки коду як «небезпечні» (unsafe). В області такого небезпечного коду не застосовуються деякі обмеження, таким чином можна виконувати операції на більш низькому рівні, але розробник повинен повністю розуміти, що він робить.

Плюси:

- безпечна робота з пам'яттю;

- висока швидкодія;
- алгебраїчний тип даних;
- передбачуваність компіляції.

Мінуси:

- деяка надмірність коду;
- висока інтенсивність розвитку мови і, як наслідок, відсутність гарної актуальної літератури для вивчення;
- необхідність чітко і однозначно прописувати параметри для компіляції [14].

3.2 Вимоги до технічного забезпечення

3.2.1 Загальні вимоги

Розроблений програмний продукт являє собою комплекс загальних функцій системи управління підсвіткою і призначений для використання будь-якою людиною.

Технічне забезпечення системи повинно максимально і найбільш ефективним чином використовувати існуючі технічні засоби.

Для коректної роботи системи управління підсвіткою до складу технічних засобів повинен входити персональний комп'ютер з такою конфігурацією:

- відеокарта з достатній об'ємом відеопам'яті (2 Гб і вище);
- процесор з тактовою частотою не нижче 1.8 ГГц;
- достатній об'єм оперативної пам'яті (2 Гб і вище);
- інші складові можуть мати довільні параметри, через те, що вони не значним чином впливають на роботу програми.

Висновки до розділу

У даному розділі було визначено програмне та технічне забезпечення для розробки системи керування підсвіткою. Були наведена основна інформація про мікроконтролер на базі Arduino Nano.

При розробці драйвера мікроконтролера було використано такі засоби для програмування на Arduino як Arduino IDE. Фактично, мова програмування пристроїв Arduino заснована на C/C++ і скомпонована з бібліотекою AVR Libc і дозволяє використовувати будь-які її функції.

Для розробки програмного забезпечення для налаштування і персоналізації фонові підсвітки використано мову програмування Rust. Описано її характеристики, а також основні переваги та недоліки.

Визначено основні вимоги до технічного забезпечення, адже технічне забезпечення системи повинно максимально і найбільш ефективним чином використовувати існуючі технічні засоби.

4 РЕАЛІЗАЦІЯ СИСТЕМИ ТА РОЗРОБКА ПРОГРАМИ

4.1 Бібліотеки мови програмування Rust для взаємодії з послідовними портами

У мові програмування Rust наявні такі бібліотеки для роботи з послідовними портами:

- serial;
- serial-core;
- serial-windows;
- serial-unix.

Кожна з них надає доступ програмам, написаним на мові програмування Rust до послідовних портів.

4.1.1 Бібліотека serial

Послідовні порти визначаються як трейти для підтримки розширення функціоналу за допомогою спеціальних реалізацій.

Трейт повідомляє компілятору Rust про функціональність певного типу та може ділитися з іншими типами. Трейти можна використовувати для того, щоб абстрактно визначити спільну поведінку. Можна використовувати рамки трейту, щоб вказати, що загальним може бути будь-який тип, який має певну поведінку.

Трейти у мові Rust схожі на функцію, яку часто називають інтерфейсами в інших мовах, хоча з деякими відмінностями.

Поведінка типу в мові програмування складається з методів, які ми можемо використовувати для цього типу. Різні типи мають однакову поведінку, якщо ми можемо викликати однакові методи для всіх цих типів. Трейти – це спосіб групування методів разом для визначення набору поведінки, необхідного для досягнення певної мети.

Наприклад, є кілька структур, які містять різні види та обсяги тексту: структура `NewsArticle`, яка містить новину, подану в певному місці, і `Tweet`, який може містити

щонайбільше 280 символів разом з метаданими, що вказує, чи це був новий твіт, ретвіт або відповідь на інший твіт.

Потрібно створити бібліотеку агрегатора медіа, яка може відображати зведення даних, які можуть зберігатися в екземплярі `NewsArticle` або `Tweet`. Для цього потрібен загальна інформації про кожний тип і в цьому допомагають трейти.

Після випуску версії бібліотеки `serial 0.4` реалізація тепер розділена на кілька бібліотек. Хоча нова організація бібліотек все ще експериментальна, бібліотека `serial` реекспортує багато типів, так що здебільшого вона сумісна із версією 0.3.

Застосування бібліотеки `serial`

- у іншій бібліотеці: повинна посилатися на бібліотеку `serial-core`;
- у файлах виконання: виконувані файли повинні вибрати реалізацію послідовного порту, крос-платформена реалізація передбачена в бібліотеці `serial`.

Сумісність

Бібліотека `serial` сумісний з Windows та будь-якою операційною системою Unix, що реалізує API `termios`. Наступні платформи підтверджені як сумісні:

- Linux (x86_64, armv6l);
- OS X (x86_64);
- FreeBSD (amd64);
- OpenBSD (amd64);
- Windows (x86_64) [15].

4.1.2 Бібліотека `serial-core`

Бібліотека `serial-core` забезпечує абстрактні типи, що використовуються для взаємодії та реалізації послідовних портів. Вони можуть бути використані для написання коду, який функціонує загально для всіх типів послідовних портів, та для реалізації нових типів послідовних портів, що працюють із загальним кодом [16].

4.1.3 Бібліотека serial-windows

Бібліотека serial-windows забезпечує послідовний порт для реалізації під операційну систему Windows.

Бібліотека serial-windows сумісна з останніми версіями операційної системи Windows. Наступні версії підтверджені як сумісні:

- Windows 7 (x86_64);
- Windows 8 (x86_64);
- Windows 10 (x86_64);

Використання

Загалом, не потрібно використовувати безпосередньо бібліотеку serial-windows. Реалізація, яку має serial-windows, також надається через крос-платформений API у бібліотеці serial.

Тип послідовного порту, визначений у serial-windows, працює з COM-портами Windows. На додаток до реалізації стандартних інтерфейсів послідовного порту, він також реалізує `std::os::windows::io::AsRawHandle`, які можна використовувати для інтеграції з іншими бібліотеками вводу-виводу [17].

4.1.4 Бібліотека serial-unix

Бібліотека serial-unix забезпечує реалізацію послідовного порту для операційних систем Unix.

Бібліотека serial-unix сумісна з будь-якою операційною системою Unix, що реалізує API `termios`. Наступні платформи підтверджені як сумісні:

- Linux (x86_64, armv6l);
- OS X (x86_64);
- FreeBSD (amd64);
- OpenBSD (amd64).

Використання

Загалом, не потрібно використовувати безпосередньо бібліотеку `serial-unix`. Реалізація, яку надає `serial-unix`, також розкривається через крос-платформний API у бібліотеці `serial`.

Тип послідовного порту, визначений у `serial-unix`, працює з будь-яким пристроєм Unix TTY. Окрім реалізації стандартних інтерфейсів послідовного порту, він також реалізує `std::os::unix::io::AsRawFd`, який можна використовувати для інтеграції з іншими бібліотеками вводу-виводу [18].

4.2 Бібліотека мови програмування Rust для захоплення зображення екрану

Scrap – бібліотека мови програмування Rust, яка призначена для захоплення зображення екрану. В цій бібліотеці містить три структури:

- Display – являє собою екран користувача;
- Frame – масив пікселів, які відображаються на екрані;
- Capturer – модуль запису.

Характеристики формату кадру:

- формат кадру гарантовано має формат BGRA;
- ширина та висота обов’язково залишаться незмінними;
- крок кадру може бути більшим за ширину, і він також може відрізнятися

залежно від кадру.

У таблиці 4.1 показано системні вимоги для роботи з бібліотекою `scrap`.

Таблиця 4.1 – Системні вимоги для роботи з бібліотекою `scrap`

Операційна система	Мінімальні вимоги
macOS	macOS 10.8
Linux	XCB + SHM + RandR
Windows	DirectX 11.1

На рисунку 4.1 показано функції, які релізовані в структурах бібліотеки `scrap` [19].

```

struct Display; /// A screen.
struct Frame; /// An array of the pixels that were on-screen.
struct Capturer; /// A recording instance.

impl Capturer {
    /// Begin recording.
    pub fn new(display: Display) -> io::Result<Capturer>;

    /// Try to get a frame.
    /// Returns WouldBlock if it's not ready yet.
    pub fn frame<'a>(&'a mut self) -> io::Result<Frame<'a>>;

    pub fn width(&self) -> usize;
    pub fn height(&self) -> usize;
}

impl Display {
    /// The primary screen.
    pub fn primary() -> io::Result<Display>;

    /// All the screens.
    pub fn all() -> io::Result<Vec<Display>>;

    pub fn width(&self) -> usize;
    pub fn height(&self) -> usize;
}

impl<'a> ops::Deref for Frame<'a> {
    /// A frame is just an array of bytes.
    type Target = [u8];
}

```

Рисунок 4.1 – Функції, які реалізовані в структурах бібліотеки `scrap`

4.3 Опис структурної та функціональної моделі

4.3.1 Опис структурної моделі

Для проектування структурної моделі системи необхідно побудувати діаграму структурну розгортання системи та діаграму структурну компонентів системи. Діаграму розгортання та діаграму компонентів наведено у Додатку А та Б відповідно.

4.3.2 Опис функціональної моделі

Для опису функціональної моделі системи для початку необхідно визначити акторів системи, а після цього визначити, які дії у системі буде виконувати кожен із акторів. Акторами у системі є користувач (людина, яка виконує налаштування), програма обробки зображення та драйвер Arduino.

UML-діаграму варіантів використання для взаємодії авторів в системі наведено у Додатку В.

4.4 Реалізація драйвера мікроконтролера

Для реалізації драйвера мікроконтролера Arduino IDE створюється файл з розширенням *.ino.

Для початку потрібно виконати налаштування. На рисунку 4.2 зображено початкові налаштування драйвера підсвітки. Потрібно налаштувати:

- кількість світлодіодів на світлодіодній стрічці, з якою буде працювати мікроконтролер;
- пін, до якого підключена стрічка;
- час (у секундах), після закінчення якого стрічка вимкнеться при попаданні сигналу;
- ліміт по струму в міліамперах (автоматично керує яскравістю): 0 ліміт вимкнено;
- параметр перевірки кольорів при запуску: 1 – увімкнено, 0 – вимкнено;
- автоматичне підлаштування яскравості від рівня зовнішнього освітлення: 1 – увімкнено, 0 – вимкнено;
- максимальну яскравість (значення від 0 до 255);
- мінімальну яскравість (значення від 0 до 255);
- константу посилення від зовнішнього світла (значення від 0 до 1023): чим менша константа, тим різкіше буде змінюватись яскравість;

– коефіцієнт фільтру (значення від 0.0 до 1.0): чим більший коефіцієнт, тим повільніше змінюється яскравість.

```
#define NUM_LEDS 98          // число світлодіодів в стрічці
#define DI_PIN 13           // пін, до якого підключена стрічка
#define OFF_TIME 10         // час (в секундах), через яке стрічка вимкнеться при пропадаанні сигналу
#define CURRENT_LIMIT 2000  // ліміт по струму в міліампер, 0 - вимкнути ліміт

#define START_FLASHES 0     // перевірка кольорів при запуску

#define AUTO_BRIGHT 1      // автоматичне підстроювання яскравості від рівня зовнішнього освітлення
#define MAX_BRIGHT 255    // максимальна яскравість (0 - 255)
#define MIN_BRIGHT 50     // мінімальна яскравість (0 - 255)
#define BRIGHT_CONSTANT 500 // константа посилення від зовнішнього світу (0 - 1023)

#define COEF 0.9            // коефіцієнт фільтру (0.0 - 1.0), чим більше - тим повільніше змінюється яскравість
```

Рисунок 4.2 – Початкові налаштування драйвера підсвітки

Для роботи підсвітки у режимі «Веселка» (плавне переливання кольорів у спектрі RGB), необхідно змінити значення константи RAINBOW (рисунок 4.3), де значення 1 – режим «Веселка» увімкнено, а значення 0 – вимкнено. У режимі «Веселка» система може працювати без підключення до персонального комп'ютера.

Константа RAINBOW_SPEED відповідає за плавність переливання, чим більше значення, тим плавніше переливаються кольори.

```
#define RAINBOW 0           // режим "Веселка" (плавне переливання кольорів), 0 - вимкнути
#define RAINBOW_SPEED 100  // швидкість переливання кольорів в режимі веселки у мілісекундах,
                          // чим більше значення - тим плавніше
```

Рисунок 4.3 – Налаштування режиму «Веселка»

Після налаштування параметрів, оголошуються змінні. На рисунку 4.4 показано приклад оголошення змінних у коді драйвера підсвітки.

Змінна `new_bright` необхідна для того, щоб зчитати показання фоторезистора.

Змінна `new_bright_f` – значення яскравості, яке ми коригуємо згідно з показаннями датчика світла.

Змінна `bright_timer` показує періодичність, з якою перевіряється, чи потрібно змінювати яскравість залежно від зовнішнього середовища.

Змінна `off_timer` показує періодичність, з якою перевіряється наявність зв'язку з персональним комп'ютером.

Константа `serialRate` показує швидкість зв'язку з персональним комп'ютером.

Масив символів `prefix[]` містить кодове слово для зв'язку з персональним комп'ютером.

`CRGB leds[NUM_LEDS]` – ініціалізація масиву діодів, таким чином, створюючи сутність стрічки.

Булева змінна `led_state` необхідна для відображення стану стрічки.

```
int new_bright, new_bright_f;
unsigned long bright_timer, off_timer;

#define serialRate 115200 // швидкість зв'язку з ПК
uint8_t prefix[] = {'A', 'd', 'a'}, hi, lo, chk, i; // кодове слово Ada для зв'язку
#include <FastLED.h>
CRGB leds[NUM_LEDS]; // створюємо об'єкт стрічки
boolean led_state = true; // флаг стану стрічки
```

Рисунок 4.4 – Оголошення змінних драйвера підсвітки

Для реалізації драйвера підсвітки також використовується бібліотека `FastLED.h`.

`FastLED.h` – бібліотека для простого та ефективного управління широким розмаїттям світлодіодних чіпсетів (таких як Neopixel, DotStar, LPD8806, WS2801). На додаток до запису на світлодіоди, ця бібліотека також включає ряд функцій для високопродуктивної 8-бітної математики для маніпулювання значеннями RGB, а також низькорівневі класи для абстрагування доступу до пінів та апаратного забезпечення SPI (англ. Serial Peripheral Interface, SPI bus – послідовний периферійний інтерфейс), зберігаючи при цьому максимальну швидкість.

При роботі з бібліотекою є певні переваги та недоліки.

Щодо переваг, то для роботи з бібліотекою потрібно підключити особисті світлодіоди, не потрібно думати про особливості використовуваних світлодіодних наборів.

Також світлодіодні чіпсети перемикаються достатньо просто: після отримання нових світлодіодів, які підтримує бібліотека, потрібно змінити визначення використовуваних світлодіодів.

Однією з переваг є також висока продуктивність – з такими функціями, як глобальне масштабування яскравості, високопродуктивна 8-бітна математика для маніпуляцій RGB та деякі найшвидші підтримки SPI, розроблені по бітах, FastLED хоче зберегти стільки циклів процесора для ваших світлодіодних моделей, скільки можливо.

Щодо недоліків, бібліотека FastLED повільніше аналога NeoPixel при будь-якій кількості світлодіодів, а інший аналог LightWS2812 показує гарну швидкість навіть на невеликій кількості світлодіодів.

Також у бібліотеки FastLED є проблеми з об'ємами пам'яті, яка нею займається. У таблиці 4.2 показано об'єми пам'яті, які використовуються бібліотекою FastLED та її аналогами (NeoPixel та LightWS2812) [20].

Таблиця 4.2 – Об'єми пам'яті, які використовуються бібліотекою FastLED та її аналогами

Пам'ять	FastLED	NeoPixel	LightWS2812
Флеш-пам'ять	12%	8%	8%
Динамічна пам'ять	25%	1%	1%

Після налаштування параметрів та додаткових змінних відбувається перехід до головних функцій програми Arduino.

Функція `setup()` виконує необхідні налаштування для роботи підсвітки. Для початку ініціює світлодіоди та передає їх у бібліотеку FastLED для подальшої роботи з діодами. Налаштовує ліміт по живленню в міліамперах, якщо такий було задано в налаштуваннях, та виконує підключення до персонального комп'ютера через послідовний порт. Функція `setup()` виконується кожний раз під час запуску контролера. На рисунку 4.5 показано функцію `setup()`.

```

void setup()
{
    FastLED.addLeds<WS2812, DI_PIN, GRB>(leds, NUM_LEDS); // ініціалізація світлодіодів
    if (CURRENT_LIMIT > 0) FastLED.setMaxPowerInVoltsAndMilliamps(5, CURRENT_LIMIT);

    // спалахи червоним синім і зеленим при запуску для перевірки роботоспроможності
    if (START_FLASHES) {
        LEDES.showColor(CRGB(255, 0, 0));
        delay(500);
        LEDES.showColor(CRGB(0, 255, 0));
        delay(500);
        LEDES.showColor(CRGB(0, 0, 255));
        delay(500);
        LEDES.showColor(CRGB(0, 0, 0));
    }

    Serial.begin(serialRate);
    Serial.print("Ada\n");
}

```

Рисунок 4.5 – Функція setup()

Функція check_connection () виконує перевірку стану підключення контролера до персонального комп'ютера. Вимикає підсвітку при відсутності сигналу через проміжок часу заданий константою OFF_TIME. На рисунку 4.6 показано функцію check_connection().

```

void check_connection() {
    if (led_state) {
        if (millis() - off_timer > (OFF_TIME * 1000)) {
            led_state = false;
            FastLED.clear();
            FastLED.show();
        }
    }
}

```

Рисунок 4.6 – Функція check_connection()

Функція loop() виконує корекцію яскравості кольорів на основі значень заданих у налаштуваннях та в залежності від зовнішнього освітлення. Зв'язується з персональним комп'ютером для отримання кольорів та відправляє отримані кольори на стрічку за допомогою бібліотеки FastLED. Функція loop() виконується у

безкінечному циклі для постійного оновлення кольорів підсвітки. На рисунку 4.8 показано функцію loop().

```
void loop() {
    if (AUTO_BRIGHT) { // якщо включена адаптивна яскравість
        if (millis() - bright_timer > 100) { // кожні 100 мс
            bright_timer = millis(); // скинути таймер
            // зчитати показання з фоторезистора, перевести діапазон
            new_bright = map(analogRead(6), 0, BRIGHT_CONSTANT, MIN_BRIGHT, MAX_BRIGHT);
            new_bright = constrain(new_bright, MIN_BRIGHT, MAX_BRIGHT);
            new_bright_f = new_bright_f * COEF + new_bright * (1 - COEF);
            LEDS.setBrightness(new_bright_f); // встановити нову яскравість
        }
    }
    if (!led_state) led_state = true;
    off_timer = millis();

    for (i = 0; i < sizeof prefix; ++i) {
waitLoop: while (!Serial.available()) check_connection();
        if (prefix[i] == Serial.read()) continue;
        i = 0;
        goto waitLoop;
    }

    while (!Serial.available()) check_connection();
    hi = Serial.read();
    while (!Serial.available()) check_connection();
    lo = Serial.read();
    while (!Serial.available()) check_connection();
    chk = Serial.read();
    if (chk != (hi ^ lo ^ 0x55))
    {
        i = 0;
        goto waitLoop;
    }

    memset(leds, 0, NUM_LEDS * sizeof(struct CRGB));
    for (uint8_t i = 0; i < NUM_LEDS; i++) {
        byte r, g, b;
        // читаем данные для каждого цвета
        while (!Serial.available()) check_connection();
        r = Serial.read();
        while (!Serial.available()) check_connection();
        g = Serial.read();
        while (!Serial.available()) check_connection();
        b = Serial.read();
        leds[i].r = r;
        leds[i].g = g;
        leds[i].b = b;
    }
    FastLED.show(); // записуємо кольору в стрічку
}
```

Рисунок 4.7 – Функція loop()

Функцію `rainbow()` генерує масив кольорів у спектрі RGB та передає кольори на світлодіодну стрічку, тим самим відтворює світловий ефект, який виглядає як веселка. У режимі «Веселка» система може працювати без персонального комп'ютера. На рисунку 4.7 показано функцію `rainbow()`.

```
void rainbow() {
  for (int j = 0; j < 255; j++) {
    for (int i = 0; i < NUM_LEDS; i++) {
      leds[i] = CHSV(i - (j * 2), MAX_BRIGHT, 255);
    }
    FastLED.show();
    delay(RAINBOW_SPEED);
  }
}
```

Рисунок 4.8 – Функція `rainbow()`

4.5 Реалізація програмного забезпечення для операційної системи

Для аналізу колірної картинки кадру на екрані і відтворення розсіяного світла по периметру оголошення використовується програмне забезпечення.

Можливе використання програми `AmbiBox` або написання власного аналогу.

4.5.1 Використання програми `AmbiBox`

За допомогою програми `AmbiBox` проводиться налаштування та персоналізація фонові підсвітки. Завдяки чому поверхня стіни за корпусом екрану динамічно висвітлюється, тим самим доповнюючи ореолом інтенсивність зображення на самому екрані і візуально збільшуючи розмір зображення.

Є кілька режимів роботи:

- захоплення екрану;
- динамічний фон;
- статичний фон;

- світломузика.

Для інтелектуальної підсвітки у програми є наступні можливості:

- різний метод захоплення, що дозволяє збільшити продуктивність і стабільність;

- гарачі клавіші на запуск підсвітки;
- гарачі клавіші на кожен профіль;
- налаштування для кожної зони та налаштування корекції кольорів;
- динамічна і статична фонові підсвітка;
- подвійний захист від перевантаження девайса від надмірної кількості даних

від великої кількості FPS (кількість кадрів в секунду);

- API для керування підсвіткою сторонніми програмами.
- плавна підсвітка;
- світломузика (крім Windows XP);
- захоплення в іграх з DirectX 9/10/11 (за допомогою PlayClaw);
- управління з пульта дистанційного керування програмою і персональним

комп'ютером;

- підтримка плагінів візуалізації для інтелектуальної підсвітки;
- немає перевантаження центрального процесора (при використанні

PlayClaw);

- є майстер налаштування зон, відмінно підходить для великої кількості зон

для таких пристроїв як Adalight;

- автоматична зміна профілю в залежності від запущеного додатку;
- підтримка 3D-відео;
- web-інтерфейс для управління інтелектуальною підсвіткою;
- підтримка Windows x32/x64 – XP/Vista/7/8/10.

- підтримка OSD;
- емуляція WinLIRC сервера;
- емуляція IRSS сервера сервера;

– запуск програм і файлів, утримуючи кнопку пульта дистанційного керування [21].

На рисунку 4.9 зображено основні налаштування підсвітки у програмі AmbiBox.

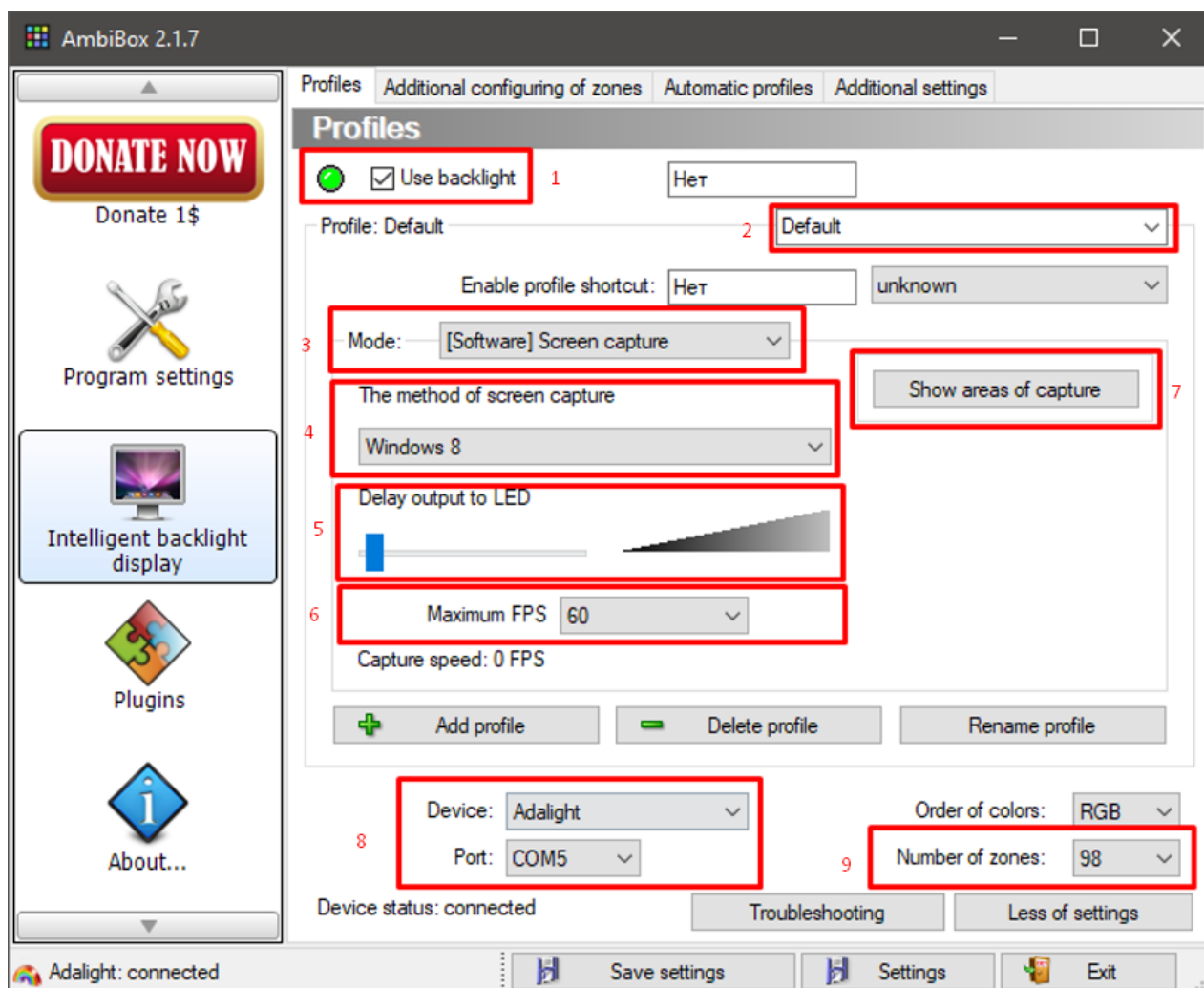


Рисунок 4.9 – Основні налаштування підсвітки

Далі наведено параметри доступні для налаштування.

- 1 – стан підсвітки, увімкнений або вимкнений;
- 2 – обраний профіль налаштувань;
- 3 – режим роботи підсвітки;
- 4 – налаштування методу захоплення зображення для режиму Screen capture;
- 5 – налаштування затримки передачі кольорів для режиму Screen capture;
- 6 – налаштування максимальної частоти кадрів для режиму Screen capture;
- 7 – перехід до налаштування зон підсвітки;
- 8 – вибір типу контролера та порту, до якого він підключений;

9 – налаштування кількості світлодіодів.

На рисунку 4.10 зображено додаткові налаштування підсвітки у програмі AmbiBox.

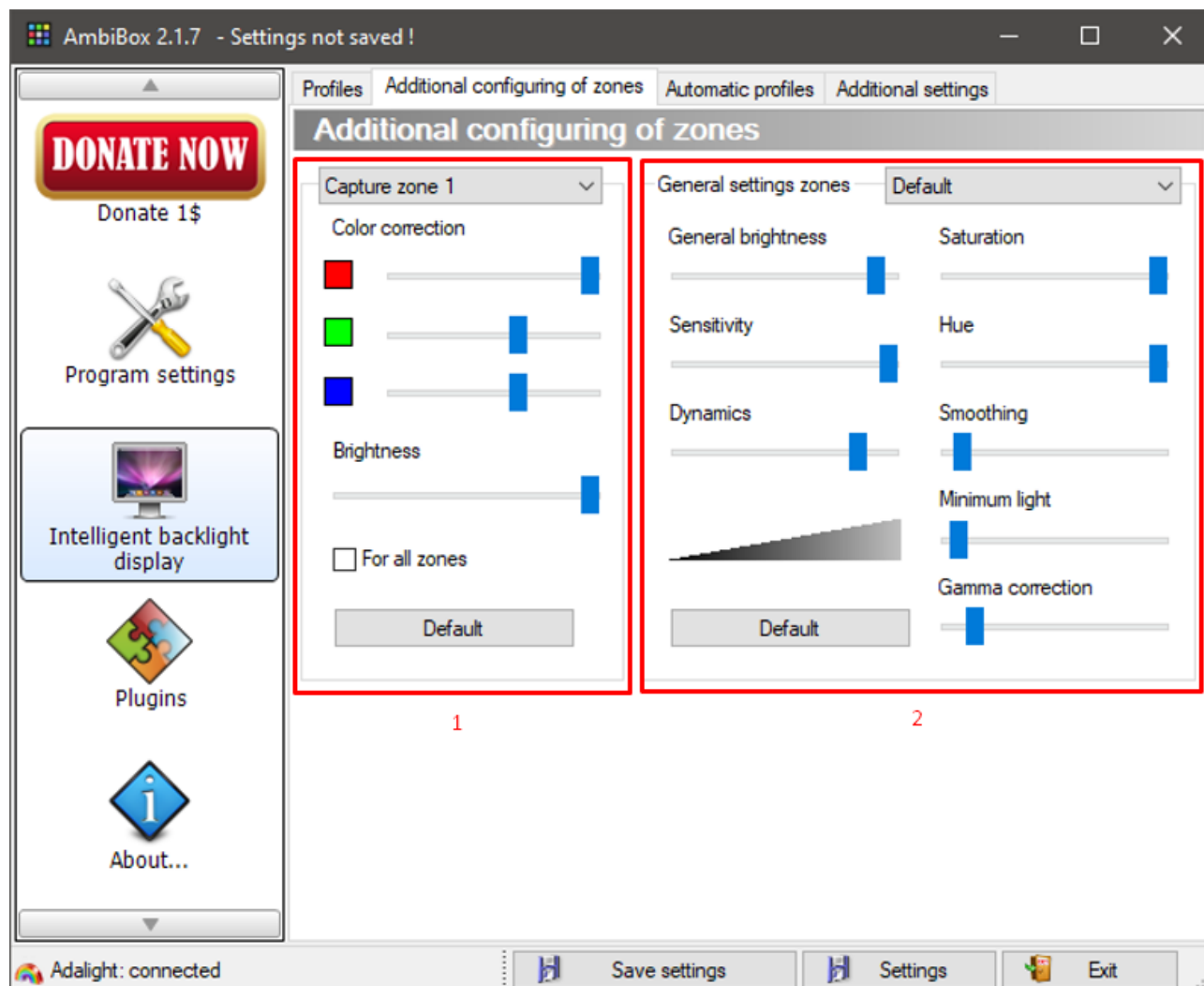


Рисунок 4.10 – Додаткові налаштування підсвітки

Далі наведено параметри доступні для налаштування.

1 – налаштування компонентів кольорів для кожної з зон;

2 – налаштування корекції кольорів, таких, як: яскравість, насиченість, згладжування, відтінок та інші.

На рисунку 4.11 зображені налаштування профілів у програмі AmbiBox.

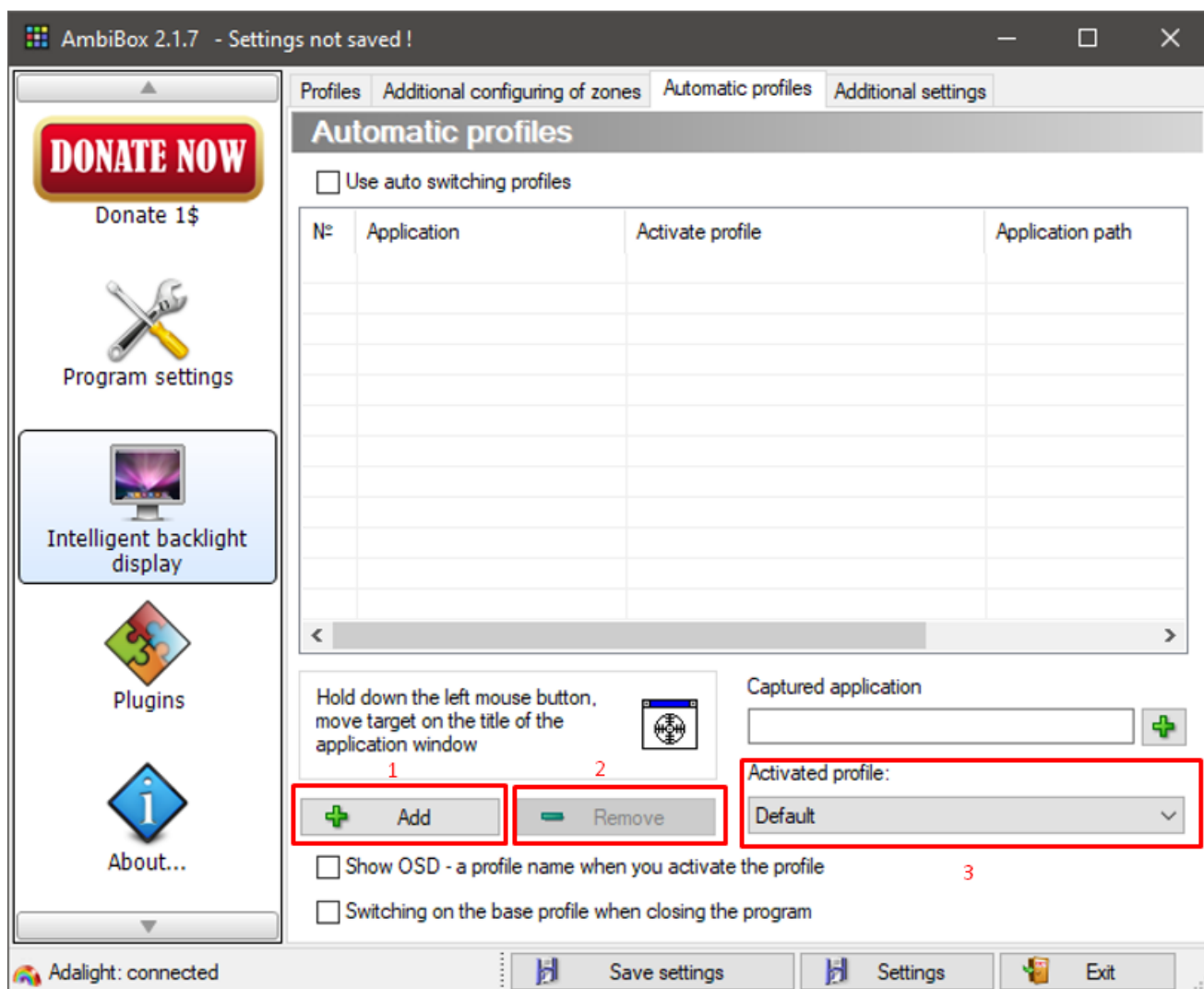


Рисунок 4.11 –Налаштування профілів підсвітки

Далі наведено параметри доступні для налаштування.

- 1 – можливість створити власний профіль налаштувань;
- 2 – можливість видалити профіль налаштувань;
- 3 – вибір профілю налаштувань.

На рисунку 4.12 зображено налаштування керування AmbiBox через API інтерфейс.

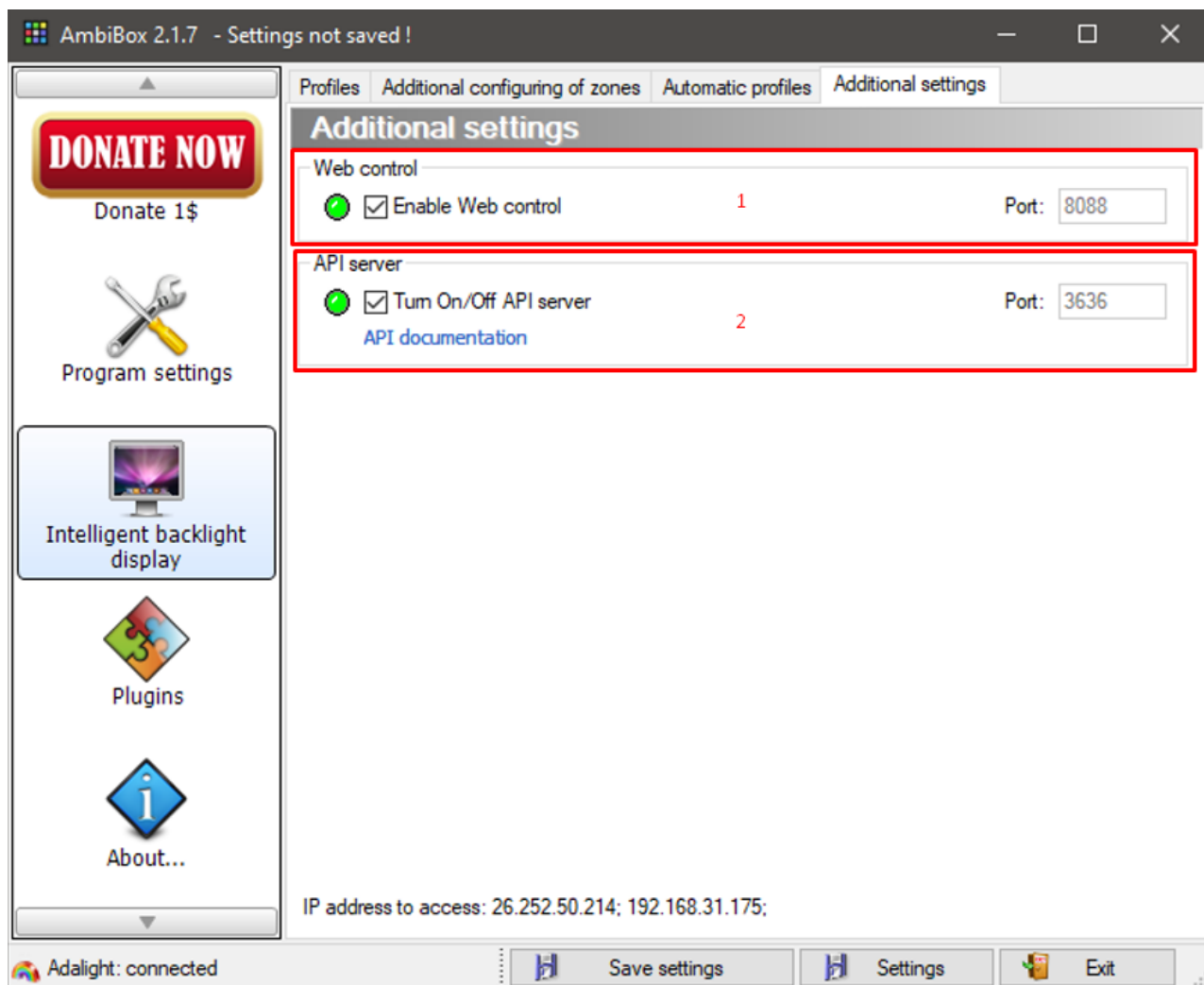


Рисунок 4.12 – Налаштування керування AmbiBox через API інтерфейс

Далі наведено параметри доступні для налаштування.

1 – стан керування через API інтерфейс, увімкнений або вимкнений.

Налаштування порту зв'язку;

2 – стан роботи серверу, увімкнений або вимкнений. Налаштування порту зв'язку.

На рисунку 4.13 зображено системні налаштування програми AmbiBox.

Далі наведено параметри доступні для налаштування.

1 – налаштування відображення вікна та іконки програми;

2 – автоматичного запуску програми при старті операційної системи, , увімкнений або вимкнений;

3 – налаштування мови графічного інтерфейсу;

4 – затримка перед увімкненням програми при старті операційної системи.

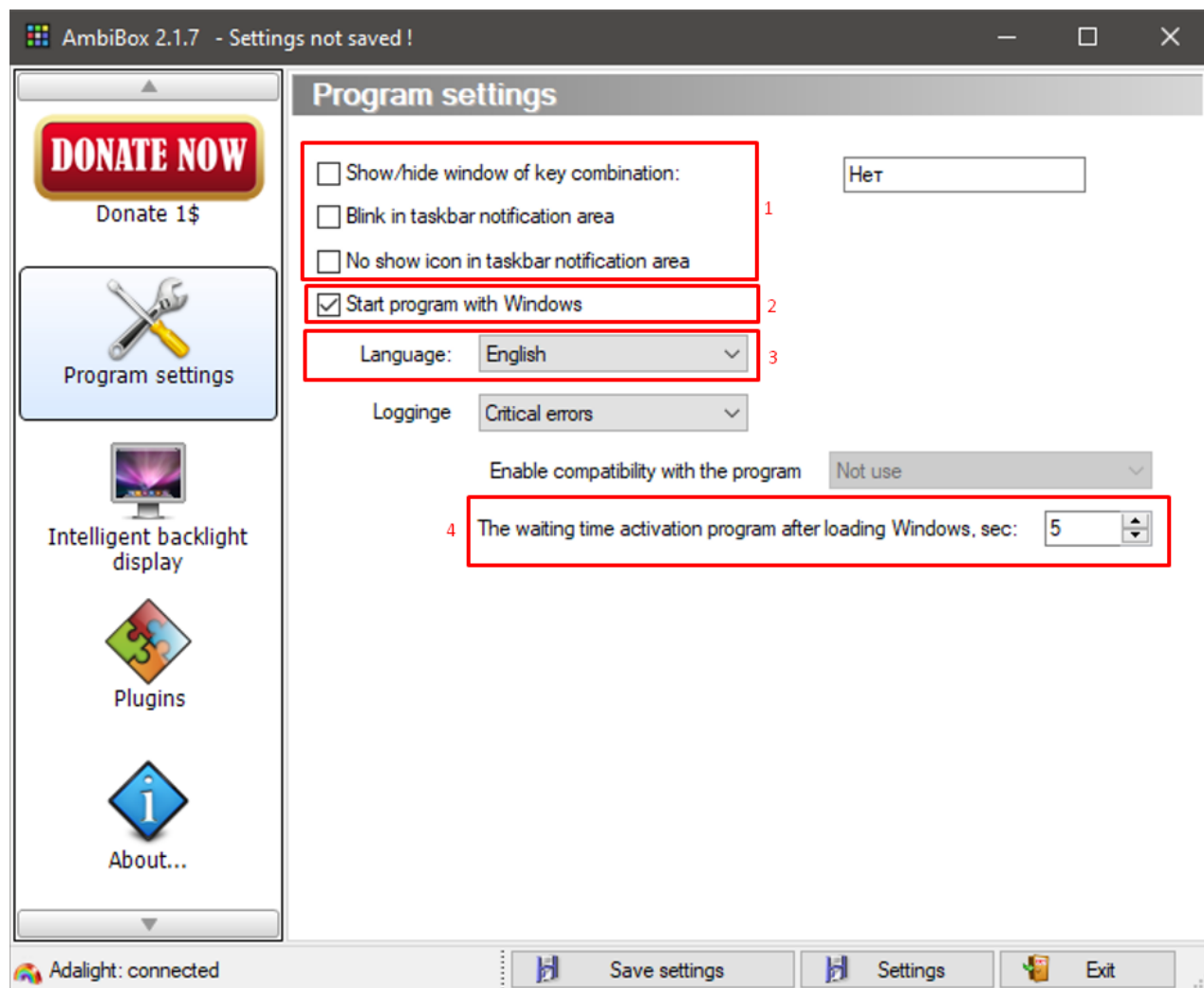


Рисунок 4.13 – Системні налаштування програми AmbiBox

Недоліки програми AmbiBox:

- не має кросплатформеності;
- при використанні перших 6 вбудованих методів захоплення, необхідно відключати Aero в Windows Vista і Windows 7 інакше Windows сама відключить Aero;
- не працює світломузика в Windows XP;
- низька частота кадрів при великій кількості зон підсвітки.

4.5.2 Реалізація програмного забезпечення

Для налаштування та персоналізації фонові підсвітки було реалізовано власне програмне забезпечення.

Для реалізації захоплення зображення екрану та роботи з послідовним портом використовуються бібліотеки `scrap` та `serial`. На рисунку 4.14 показано імпорт бібліотек та модулів.

```
extern crate scrap;

use clap::{App, Arg};
use scrap::{Capturer, Display};
use serial::prelude::*;
use std::io;
use std::thread;
use std::time::Duration;
use time;
```

Рисунок 4.14 – Імпорт бібліотек та модулів

Структура `Pixel` (рисунок 4.15) має поля `r`, `g`, `b` для зберігання компонентів кольорів кожного пікселя.

Структура `Point` (рисунок 4.15) має поля `x` та `y` для зберігання координат розташування точки.

```
struct Pixel {
    r: u8,
    g: u8,
    b: u8,
}

struct Point {
    x: u32,
    y: u32,
}
```

Рисунок 4.15 – Основні структури програми

Структура `Config` (рисунок 4.16) необхідна для зберігання даних налаштувань. Поля структури `Config` відповідають полям файлу `config.json`.

Схема зон підсвітки наведена у Додатку Д.

```
struct Config {  
    display_index: usize,  
    horizontal_screen_size: usize,  
    vertical_screen_size: usize,  
    horizontal_leds_count: u32,  
    vertical_leds_count: u32,  
    horizontal_area_height: u32,  
    vertical_area_width: u32,  
    padding_top: u32,  
    padding_bottom: u32,  
    padding_left: u32,  
    padding_right: u32,  
    port: usize,  
}
```

Рисунок 4.16 – Структура налаштування програми

Функція `read_config_file` (рисунок 4.17) виконує зчитування файлу налаштувань та приведення тексту до об'єкту типу `Config` для подальшого використання даних налаштувань у програмі.

```
fn read_config_file() -> Config {  
    let mut file = File::open("config.json").unwrap();  
    let mut buff = String::new();  
    file.read_to_string(&mut buff).unwrap();  
  
    let config: Config = serde_json::from_str(&buff).unwrap();  
    return config;  
}
```

Рисунок 4.17 – Функція для зчитування файлу налаштувань

Функція `get_pixel` (рисунок 4.18) отримує масив усіх кольорів на екрані, вираховує необхідні значення та повертає об'єкт типу `Pixel`.

```
fn get_pixel(frame: &std::vec::Vec<dxgcap::BGRA8>,
             x: usize, y: usize,
             screen_height: usize
            ) -> Pixel {
    let stride = frame.len() / screen_height;
    let start = stride * y + x;
    Pixel {
        r: frame[start].r,
        g: frame[start].g,
        b: frame[start].b,
    }
}
```

Рисунок 4.18 – Функція яка повертає піксель

Функція `prepare_port` (рисунок 4.19) виконує конфігурацію необхідних параметрів послідовного порту для обміну інформацією з мікроконтролером.

```
fn prepare_port<T: SerialPort>(port: &mut T) -> io::Result<()> {
    port.reconfigure(&|settings| {
        settings.set_baud_rate(serial::Baud115200)?;
        settings.set_char_size(serial::Bits8);
        settings.set_parity(serial::ParityNone);
        settings.set_stop_bits(serial::Stop1);
        settings.set_flow_control(serial::FlowNone);
        Ok(())
    })?;
    port.set_timeout(Duration::from_millis(5))?;
    std::thread::sleep(Duration::from_secs(4));
    return Ok(());
}
```

Рисунок 4.19 – Функція конфігурації послідовного порту

Функція `main` (рисунок 4.20) являє собою головний функцію програми, яка виконується під час її запуску. В ній виконуються налаштування базових параметрів,

формування шляху до послідовного порту, ініціалізація дисплею а також обрахування необхідних змінних для роботи програми.

```
fn main() {
    let matches = App::new("Advertising light driver")
        .version("0.1.0")
        .author("Koshoviy Timur")
        .about("System for controlling advertising lighting")
        .arg(
            Arg::with_name("port")
                .short("p")
                .long("port")
                .takes_value(true)
                .help("Serial port connected to device"),
        )
        .arg(
            Arg::with_name("debug")
                .short("d")
                .help("turns on debugging mode"),
        )
        .get_matches();
    let conf: Config = read_config_file();
    let port_abr = String::from("COM");
    let mut port_str: String = format!("{}", port_abr, &conf.port);
    if cfg!(unix) {
        let port_abr = String::from("/dev/ttyUSB");
        port_str = format!("{}", port_abr, &conf.port);
    }
    let port_path = matches.value_of("port").unwrap_or(&port_str);

    let debug_mode = matches.occurrences_of("debug") > 0;

    let one_second = Duration::new(1, 0);
    let one_frame = one_second / 60;

    let mut display = Capturer::new(conf.display_index).unwrap();
    let (w, h) = (conf.horizontal_screen_size, conf.vertical_screen_size);
    let mut port = serial::open(port_path).unwrap();
    prepare_port(&mut port).expect("Couldn't configure serial port");
}
```

Рисунок 4.20 – Головний метод програми

На рисунку 4.21 показано обрахування змінних, які необхідні для визначення кольорів зон.

```
let start_up_point = Point {  
  x: conf.padding_left,  
  y: conf.padding_top,  
};  
  
let end_up_point = Point {  
  x: w as u32 - conf.padding_right,  
  y: conf.padding_top + conf.horizontal_area_height,  
};  
  
let start_down_point = Point {  
  x: conf.padding_left,  
  y: h as u32 - conf.padding_bottom - conf.horizontal_area_height,  
};  
  
let end_down_point = Point {  
  x: w as u32 - conf.padding_right,  
  y: h as u32 - conf.padding_bottom,  
};  
  
let left_up_point = Point {  
  x: conf.padding_left,  
  y: conf.padding_top,  
};  
  
let left_down_point = Point {  
  x: conf.padding_left + conf.vertical_area_width,  
  y: h as u32 - conf.padding_bottom,  
};  
  
let right_up_point = Point {  
  x: w as u32 - conf.padding_right - conf.vertical_area_width,  
  y: conf.padding_top,  
};  
let right_down_point = Point {  
  x: w as u32 - conf.padding_right,  
  y: h as u32 - conf.padding_bottom,  
};
```

Рисунок 4.21 – Змінні для циклу loop

На рисунку 4.22 показано цикл в кожній ітерації якого виконується захоплення зображення з екрану, виклик функцій для обрахування необхідних кольорів, приведення кольорів у необхідний формат та відправка результату на послідовний порт.

```
loop {
    let frame_time_start = time::precise_time_ns();

    let buffer = match display.capture_frame() {
        Ok(buffer) => buffer,
        Err(error) => {
            println!("SLEEPING");
            thread::sleep(one_frame);
            continue;
        }
    };

    let mut lower = count_horizontal(&buffer, &start_down_point, &end_down_point, &conf);
    lower.reverse();
    let mut left = count_vertical(&buffer, &left_up_point, &left_down_point, &conf);
    left.reverse();
    let mut upper = count_horizontal(&buffer, &start_up_point, &end_up_point, &conf);
    let mut right = count_vertical(&buffer, &right_up_point, &right_down_point, &conf);

    let data: Vec<Pixel> = [
        lower,
        left,
        upper,
        right,
    ]
    .concat();
    send_vec(&mut port, &data).expect("Failed to send data to ambilight device");
    let frame_time_end = time::precise_time_ns();
    if debug_mode {
        println!(
            "frame time elapsed {}ms",
            (frame_time_end - frame_time_start) / 1000000
        );
    }
}
```

Рисунок 4.22 – Цикл loop

На рисунку 4.23 показано функцію, яка встановлює з'єднання з мікроконтролером через послідовний порт та відправляє на контролер необхідні для відображення кольори.

```
fn send_vec<T: SerialPort>(port: &mut T, buf: &[Pixel]) -> io::Result<()> {
    let ada = [65, 100, 97];
    let check = [5, 6, 86];
    port.write(&ada)?;
    port.write(&check)?;
    for p in buf.iter() {
        port.write(&[p.r, p.g, p.b])?;
    }
    Ok(())
}
```

Рисунок 4.23 – Функція відправки кольорів на мікроконтролер

На рисунках 4.24 - 4.25 показано функції `count_vertical` та `count_horizontal`. Дані функції отримують масив пікселів екрану та виконують обрахування кольорів для вертикальних та горизонтальних зон підсвітки відповідно. Обрахування виконується на основі заданих у файлі `config.json` налаштувань.

```
fn count_vertical(frame: &std::vec::Vec<dxgcap::BGRA8>,
start_point: &Point,
end_point: &Point,
conf: &Config
) -> Vec<Pixel> {
    let area_height = (end_point.y - start_point.y) / conf.vertical_leds_count;
    let mut result: Vec<Pixel> = Vec::with_capacity(conf.vertical_leds_count as usize);
    for i in 0..conf.vertical_leds_count {
        let start_y = start_point.y + area_height * i as u32;
        let end_y = start_y + area_height;
        let mut r: u64 = 0;
        let mut g: u64 = 0;
        let mut b: u64 = 0;
        for y in (start_y..end_y).step_by(STEP_SIZE) {
            for x in (start_point.x..end_point.x).step_by(STEP_SIZE) {
                let pixel = get_pixel(&frame, x as usize, y as usize, conf.vertical_screen_size);
                r += pixel.r as u64;
                g += pixel.g as u64;
                b += pixel.b as u64;
            }
        }
        let area: u64 = (start_y..end_y).step_by(STEP_SIZE).count() as u64
            * (start_point.x..end_point.x).step_by(STEP_SIZE).count() as u64;
        r = r / area;
        g = g / area;
        b = b / area;
        let p = Pixel {
            r: r as u8,
            g: g as u8,
            b: b as u8,
        };
        result.push(p);
    }
    return result;
}
```

Рисунок 4.23 – Функція яка повертає масив вертикальних кольорів

```

fn count_horizontal(
    frame: &std::vec::Vec<dxgcap::BGRA8>,
    start_point: &Point,
    end_point: &Point,
    conf: &Config
) -> Vec<Pixel> {
    let horizontal_area_width = (end_point.x - start_point.x) / conf.horizontal_leds_count as u32;
    let mut result: Vec<Pixel> = Vec::with_capacity(conf.horizontal_leds_count as usize);
    for i in 0..conf.horizontal_leds_count {
        let start_x = start_point.x + horizontal_area_width * i as u32;
        let end_x = start_x + horizontal_area_width;
        let start_y = start_point.y;
        let end_y = start_y + conf.horizontal_area_height;
        let mut r: u64 = 0;
        let mut g: u64 = 0;
        let mut b: u64 = 0;
        for x in (start_x..end_x).step_by(STEP_SIZE) {
            for y in (start_y..end_y).step_by(STEP_SIZE) {
                let pixel = get_pixel(&frame, x as usize, y as usize, conf.vertical_screen_size);
                r += pixel.r as u64;
                g += pixel.g as u64;
                b += pixel.b as u64;
            }
        }
        let area: u64 = (start_y..end_y).step_by(STEP_SIZE).count() as u64
            * (start_point.x..end_point.x).step_by(STEP_SIZE).count() as u64;
        r = r / area;
        g = g / area;
        b = b / area;
        let p = Pixel {
            r: r as u8,
            g: g as u8,
            b: b as u8,
        };
        result.push(p);
    }
    result
}

```

Рисунок 4.25– Функція яка повертає масив горизонтальних кольорів

Функція `prepare_port` (рисунок 4.24) виконує конфігурацію необхідних параметрів послідовного порту для обміну інформацією з мікроконтролером.

4.6 Керівництво користувача

4.6.1 Налаштування мікроконтролера Arduino

Для початку необхідно відкрити файл з розширенням `*.ino` в Arduino IDE та виконати налаштування.

Визначаємо кількість світлодіодів на світлодіодній стрічці, якою буде керувати система. Потрібно порахувати та присвоїти це число константі NUM_LEDS, в даному випадку кількість світлодіодів дорівнює 98.

Після цього потрібно вказати до якого піну мікроконтролера на базі Arduino підключений контакт цифрового входу (англ. DI – digital input) світлодіодної стрічки, в даному випадку DI_PIN = 13.

Також на випадок зникнення цифрового сигналу, встановлюється таймер, після закінчення якого підсвітка вимкнеться, якщо сигнал не відновиться.

Зі збільшеннями кількості світлодіодів зростає і енергоспоживання, але можна примусово встановити ліміт на максимальне споживання енергії. Система буде автоматично підлаштовувати яскравість для того, щоб не вийти за рамки заданого ліміту. При встановленні ліміту в 0 – він вважається вимкненим.

Якщо виявляється проблема в роботі системи, то параметру START_FLASHES можна присвоїти значення 1 для того, щоб при запуску системи переконатися у працездатності світлодіодної стрічки. Перевірка відбувається наступним чином: стрічка по чергові засвітиться червоним, синім та зеленим кольорами. Якщо проблем немає – стрічка переходить у свій звичайний режим, інакше стрічка гасне.

Для того, щоб обмежити максимальну яскравість підсвітки, потрібно задати значення константи MAX_BRIGHT, де 0 – це мінімальне значення, а 255 – максимальне. Аналогічним чином налаштовується мінімальна яскравість.

Константа AUTO_BRIGHT вмикає автоматичне налаштування яскравості. Якщо до мікроконтролеру не під'єднано фоторезистора, то це значення ніяк не впливає на роботу підсвітки.

Якщо фоторезистор все ж під'єднаний до мікроконтролера, то бажано налаштувати константу посилення від зовнішнього світла, де її мінімальне значення це – 0, а максимальне це – 1023. Чим більше значення константи, тим плавніше збільшується яскравість підсвітки при збільшенні освітлення від зовнішніх джерел. В даному випадку було взято середнє значення – 500.

Остання константа COEF відповідає за швидкість зміни яскравості. Може приймати значення від 0 до 1.0. Чим більше значення, тим більше часу займає зміна яскравості.

На рисунку 4.26 зображено фрагмент коду з константами драйвера мікроконтролера.

```
#define NUM_LEDS 98
#define DI_PIN 13
#define OFF_TIME 10
#define CURRENT_LIMIT 2000

#define START_FLASHES 0

#define AUTO_BRIGHT 1
#define MAX_BRIGHT 255
#define MIN_BRIGHT 50
#define BRIGHT_CONSTANT 500

#define COEF 0.9
```

Рисунок 4.26 – Фрагмент коду з константами драйвера мікроконтролера

Після присвоєння значень констант, необхідно під'єднати плату Arduino до персонального комп'ютера за допомогою USB кабелю та визначити номер порту. Для цього потрібно на комп'ютері на базі операційної системи Windows перейти до програми «Диспетчер пристроїв», потім натиснути на вкладку «Порти» та знайти порт, до якого підключено плату. Оскільки плата Arduino Nano має на своєму борті перетворювач CH340, необхідно знайти у списку пристрій, який має CH340 у своїй назві. На рисунку 4.30 наведено приклад відображення плати Arduino Nano у диспетчері пристроїв.

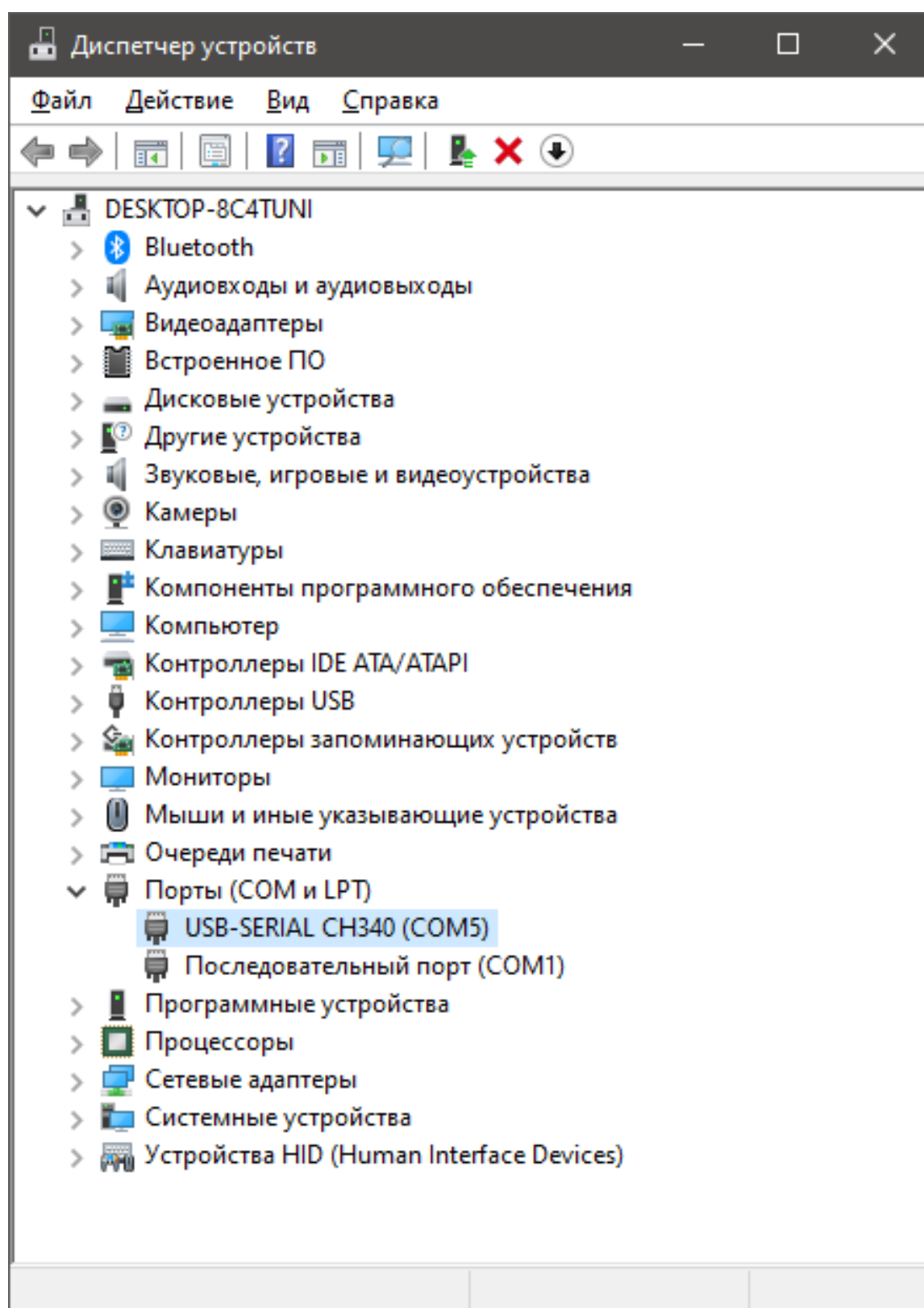


Рисунок 4.27 – Приклад відображення плати Arduino Nano у диспетчері пристроїв

Наступним кроком буде завантаження драйверу на пам'ять мікроконтролера.

Для початку в Arduino IDE потрібно натиснути на кнопку «Інструменти» та вибрати плату (рисунок 4.28).

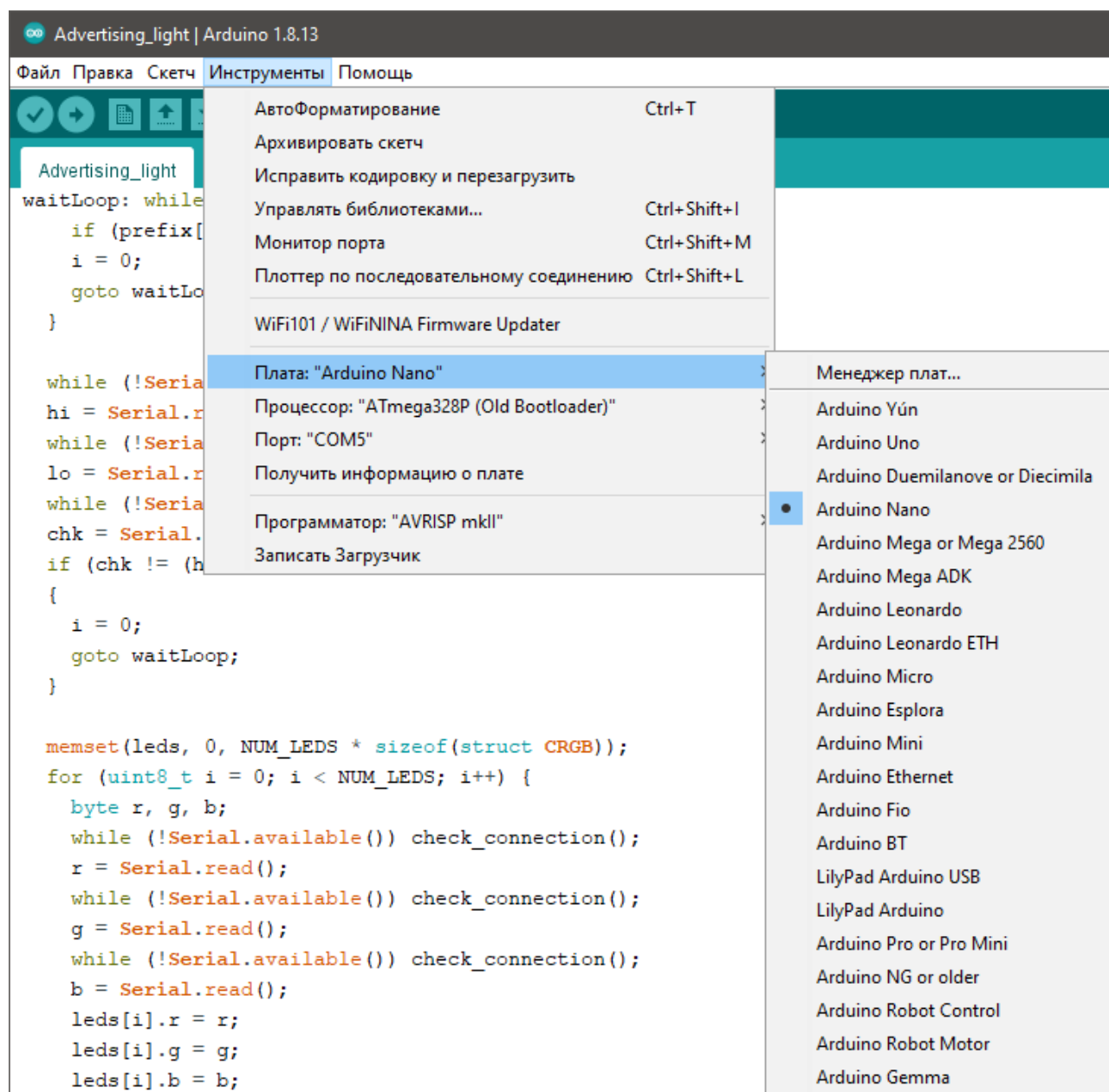


Рисунок 4.28 – Вибір плати Arduino Nano

Далі обираємо порт, до якого під'єднано мікроконтролер (рисунок 4.29).

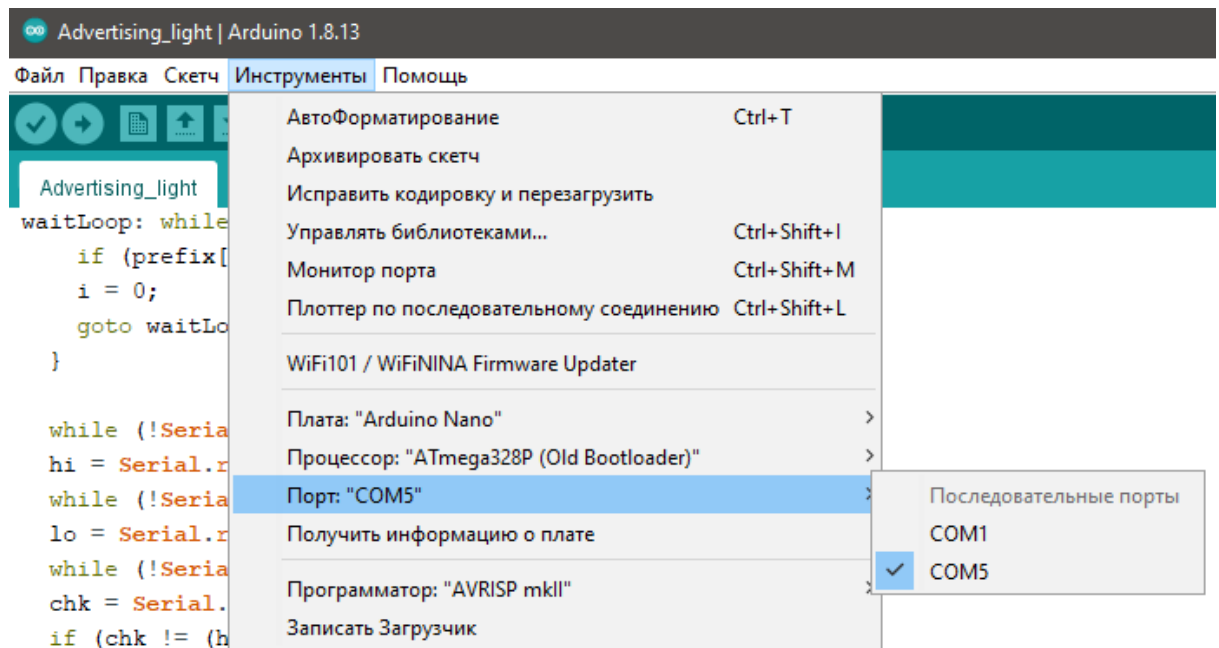


Рисунок 4.29 – Вибір порту COM5

І останнім виконується компіляція і завантаження драйвера в пам'ять мікроконтролеру (рисунок 4.30).

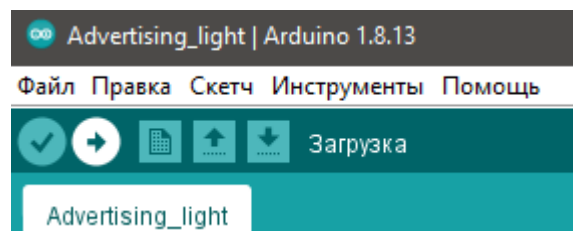


Рисунок 4.30 – Компіляція і завантаження драйвера мікроконтролера

На рисунку 4.31 показано повідомлення про успішне завантаження драйверу на пам'ять мікроконтролера.

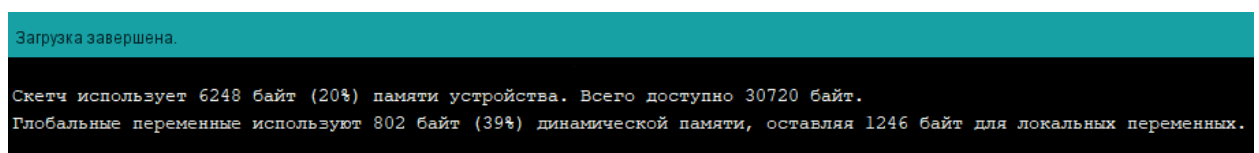


Рисунок 4.31 – Компіляція і завантаження драйвера мікроконтролера

4.6.2 Налаштування AmbiBox

При використанні AmbiBox замість власного програмного забезпечення, потрібно від час встановлення програми AmbiBox вибрати пристрій Adalight (рисунок 4.32)

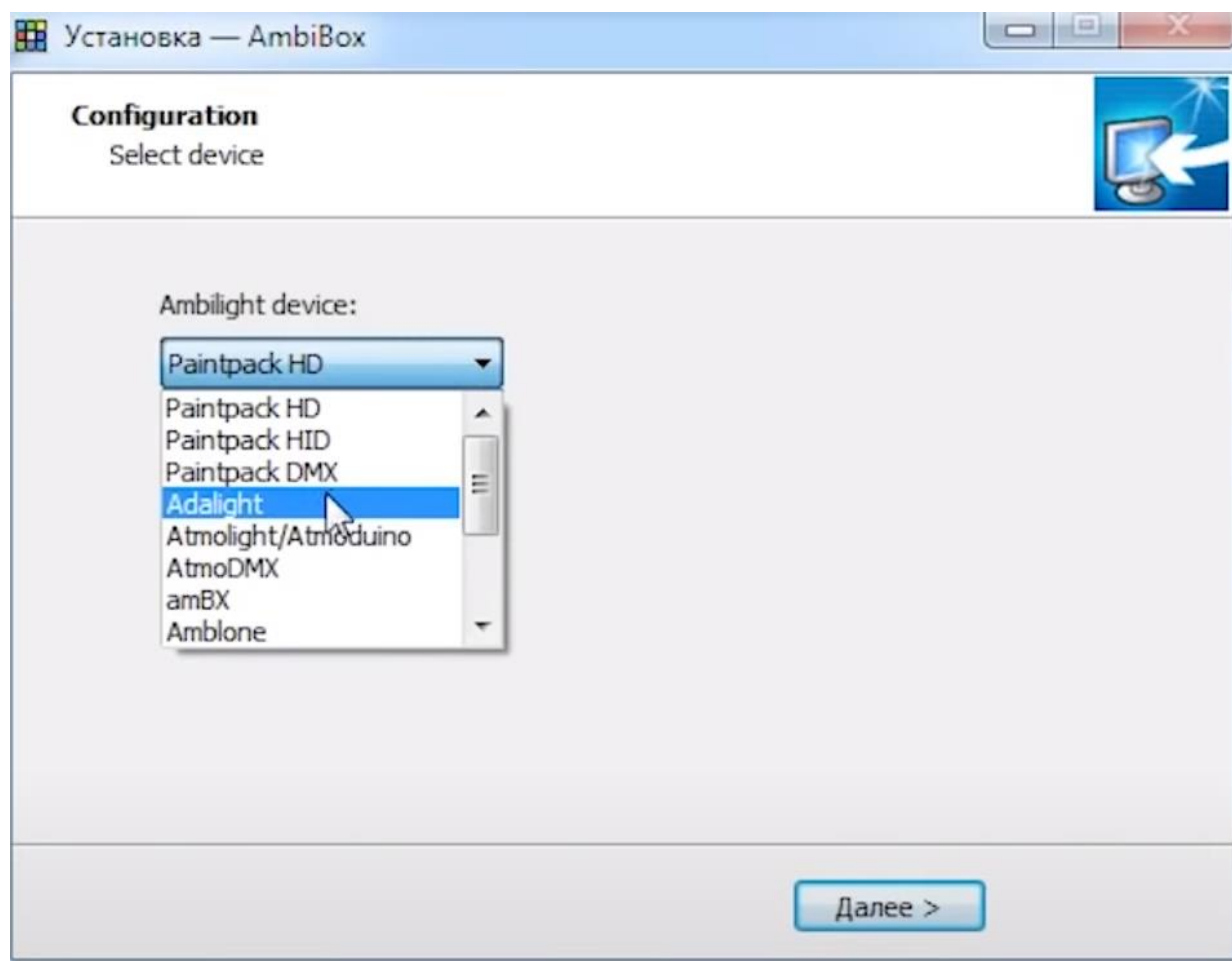


Рисунок 4.32 – Вибір пристрою Adalight під час встановлення

Далі необхідно налаштувати програму AmbiBox. По перше, потрібно натиснути кнопку «More settings» для того, щоб відкрити доступ до більшої кількості налаштувань (рисунок 4.33).

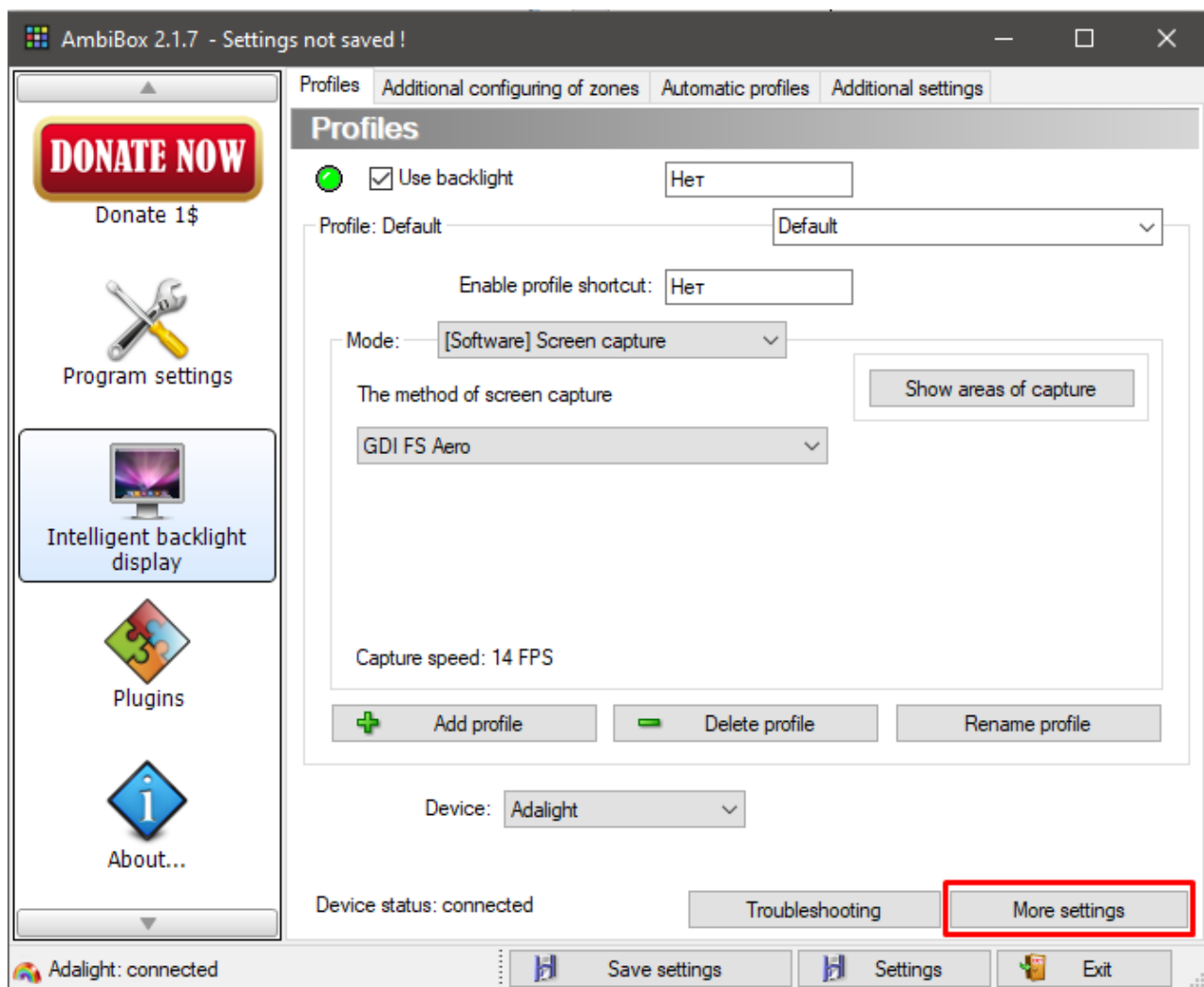


Рисунок 4.33 – Кнопка «More settings»

Далі наведено параметри, які потрібно налаштувати (рисунок 4.34).

- 1 – переконатися, що присутня галочка в чекбоксі «User backlight»;
- 2 – обрати режим роботи підсвітки: «Screen capture»;
- 3 – переконатися, що вибрано пристрій «Adalight»;
- 4 – обрати порт до якого підключено мікроконтролер;
- 5 – вибрати таку ж кількість світлодіодів, як і в драйвері мікроконтролера.

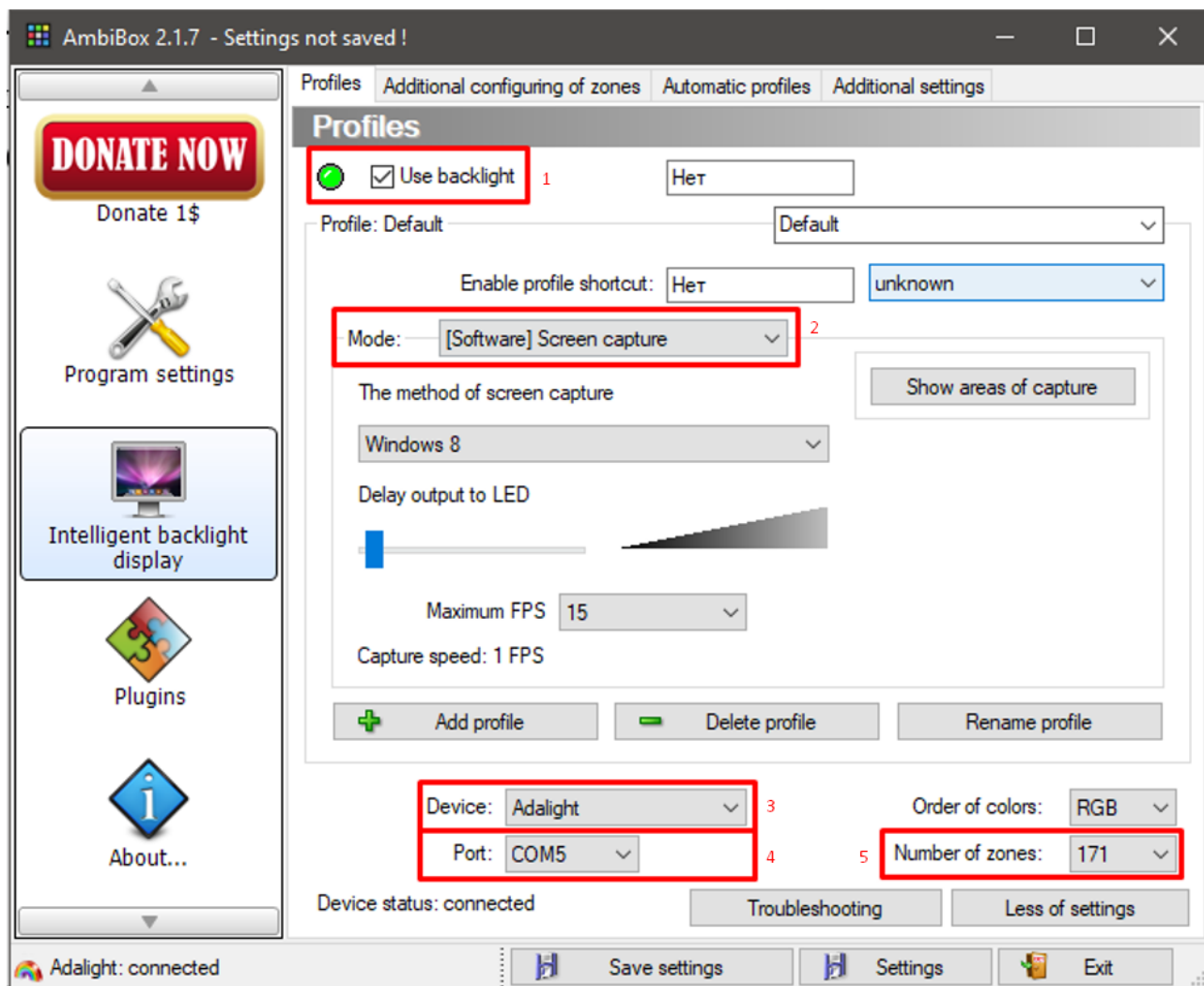


Рисунок 4.34 – Налаштування параметрів

Далі необхідно налаштувати зони, для цього потрібно натиснути на кнопку «Show areas of capture», а потім на кнопку «Wizard capture zones». У вікні що відкрилося (рисунок 4.35), необхідно налаштувати параметри.

- 1 – кількість світлодіодів по горизонталі;
- 2 – кількість світлодіодів по вертикалі;
- 3 – співвідношення сторін;
- 4 – розмір зон для вирахування кольорів.

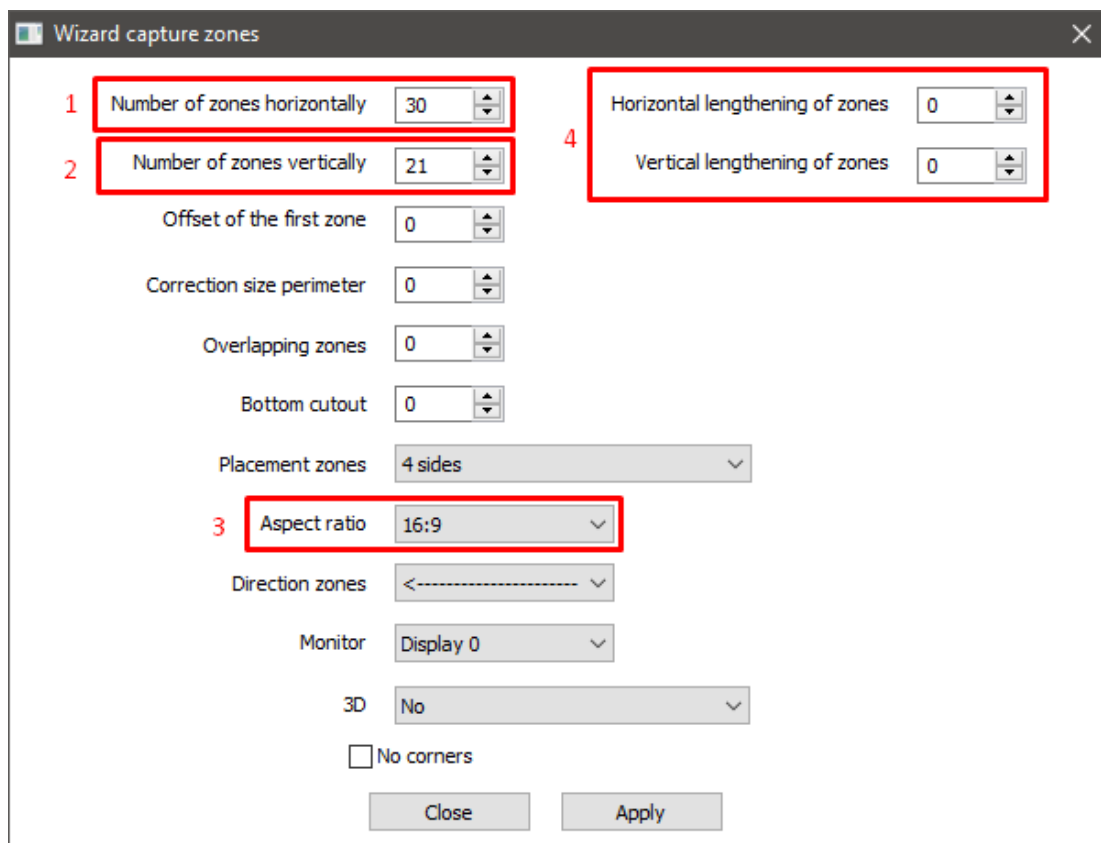


Рисунок 4.35 – Налаштування параметрів зон

4.6.3 Налаштування власного програмного забезпечення

Для використання власного програмного забезпечення необхідно відкрити файл налаштувань, який називається config.json (рисунок 4.36).

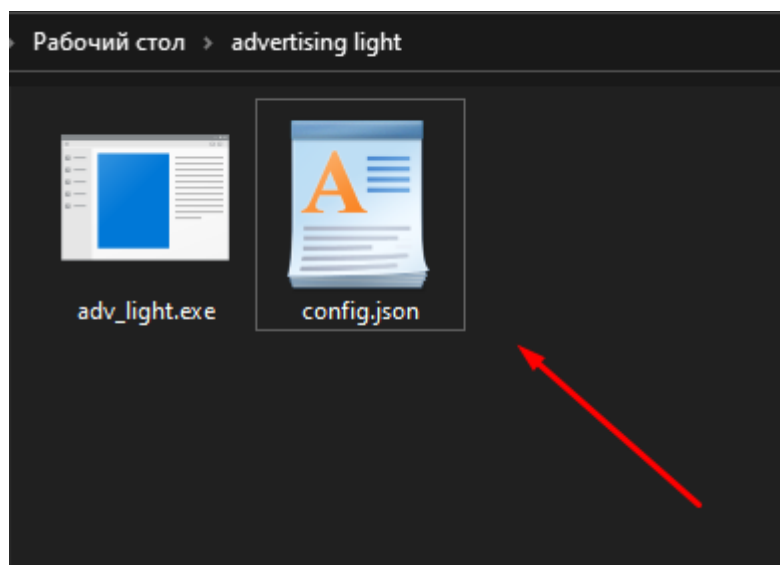


Рисунок 4.36 – Файл налаштувань

Далі у файлі config.json необхідно налаштувати наступні параметри (рисунок 4.37).

```

1  {
2      "display_index": 0,
3      "horizontal_screen_size": 2560,
4          "vertical_screen_size": 1440,
5          "horizontal_leds_count": 28,
6          "vertical_leds_count": 16,
7          "horizontal_area_height": 400,
8          "vertical_area_width": 400,
9          "padding_top": 100,
10         "padding_bottom": 100,
11         "padding_left": 50,
12         "padding_right": 50,
13         "port": 5
14     }

```

Рисунок 4.37 – Параметри налаштувань

`display_index` – індекс монітору або екрану, з якого буде захоплюватися зображення. Якщо до персонального комп'ютера підключено лише один екран, то необхідно поставити в даному полі значення 0. Якщо підключено більше ніж один монітор або екран, то необхідно визначити індекс монітору з якого буде захоплюватися зображення.

`horizontal_screen_size` – кількість пікселів монітору або екрану по горизонталі.

`vertical_screen_size` – кількість пікселів монітору або екрану по вертикалі.

`horizontal_leds_count` – кількість світлодіодів на стрічці по горизонталі.

`vertical_leds_count` – кількість світлодіодів на стрічці по вертикалі.

`horizontal_area_height` – висота для горизонтальних зон світлодіодів у пікселях.

`vertical_area_width` – ширина для вертикальних зон світлодіодів у пікселях.

`padding_top` – відступ від верхнього краю зображення для верхніх зон у пікселях.

`padding_bottom` – відступ від краю зображення для нижніх зон у пікселях.

`padding_left` – відступ від краю зображення для зон, які знаходяться зліва у пікселях.

`padding_right` – відступ від краю зображення для зон, які знаходяться справа у пікселях.

`port` – номер порту, до якого підключено мікроконтролер.

Після цього необхідно зберегти введені налаштування та запустити файл `adv_light.exe` (рисунок 4.38).

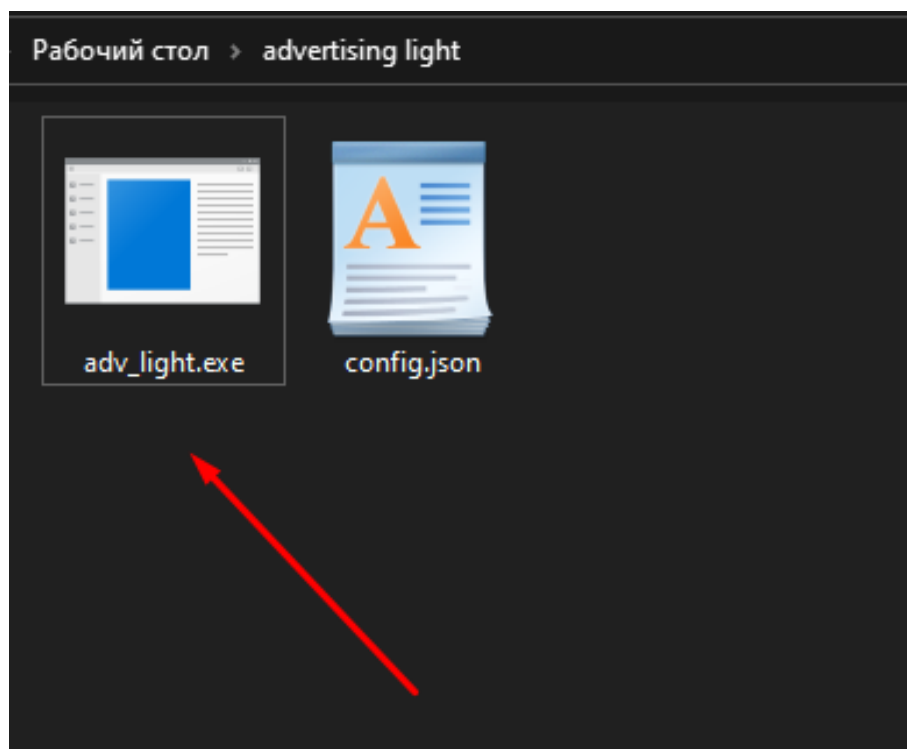


Рисунок 4.38 – Файл запуску підсвітки

4.6.4 Робота підсвітки

Після успішних налаштувань драйвера мікроконтролера та власного програмного забезпечення або AmbiBox, користувач отримає наступні результати. На рисунках 4.39 – 4.42 показано роботу системи на прикладі домашнього монітору.



Рисунок 4.39 – Результаты работы системы



Рисунок 4.40 – Результаты работы системы

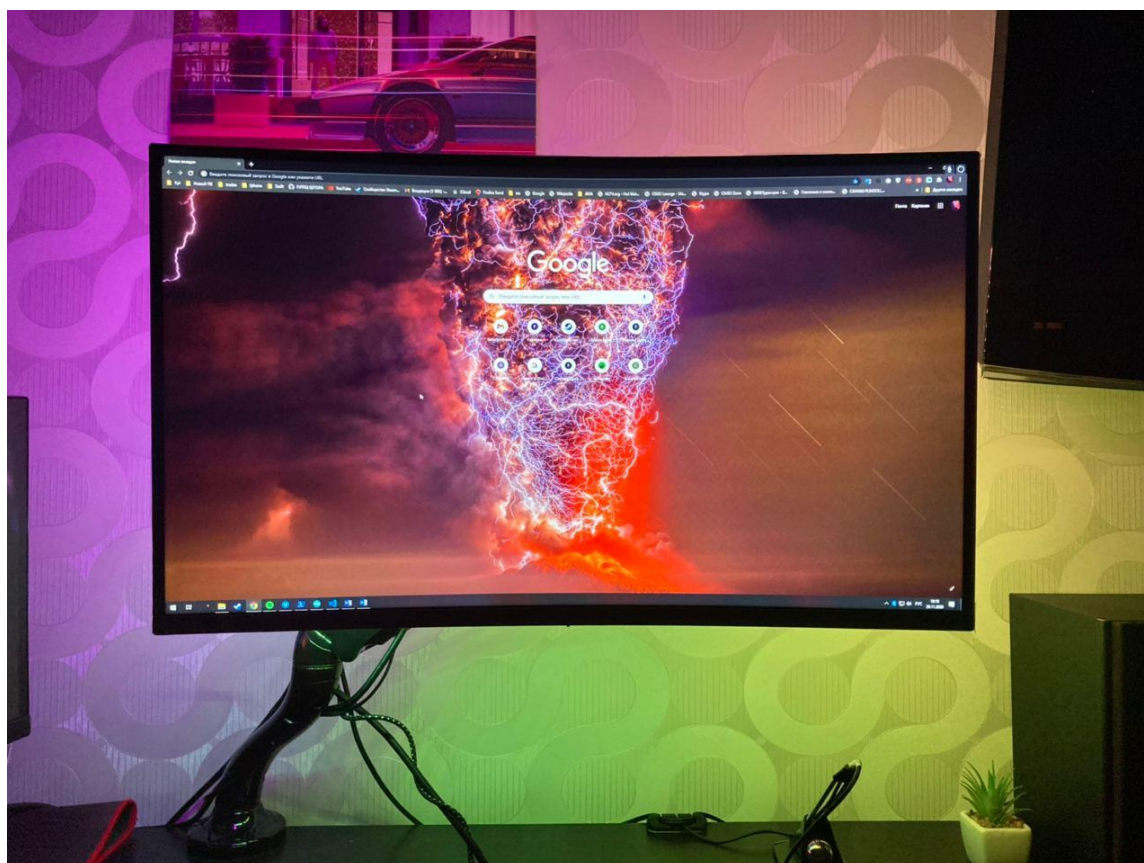


Рисунок 4.41 – Результаты работы системы в режиме «Веселка»

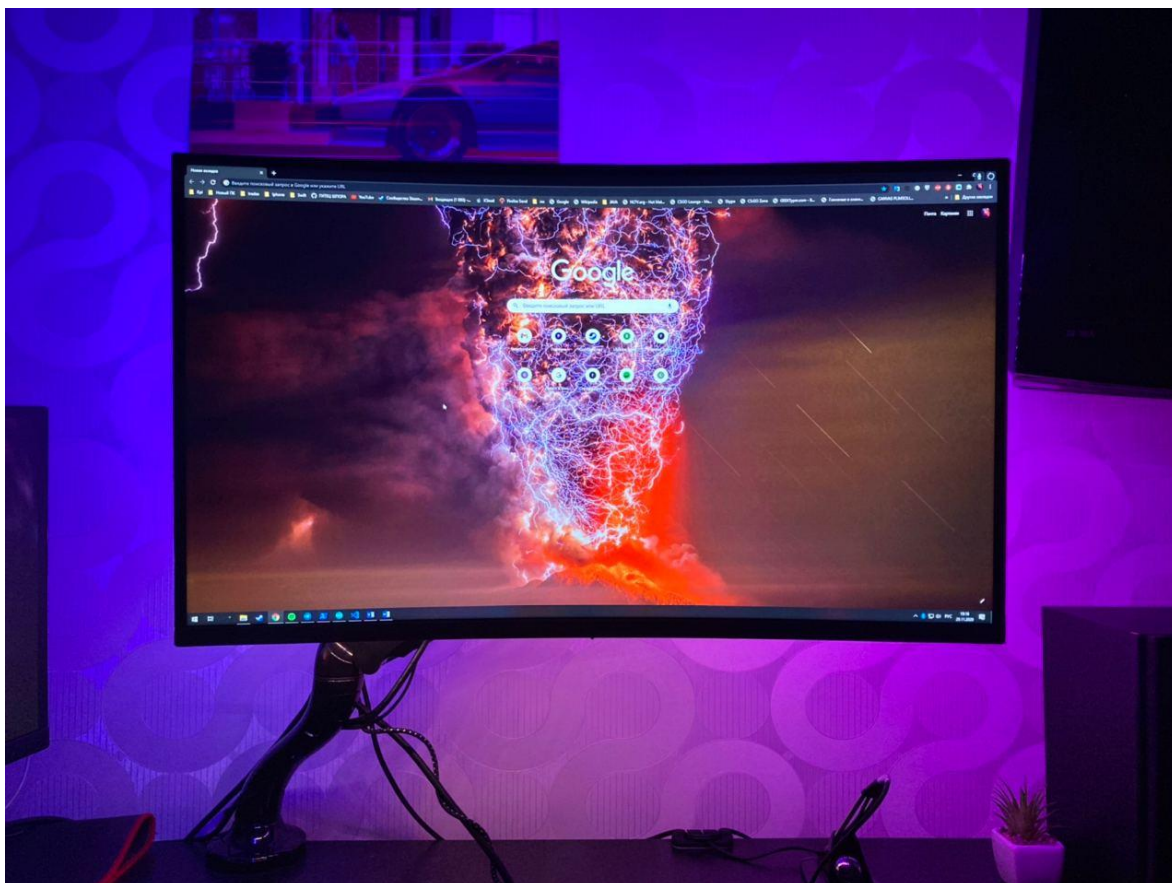


Рисунок 4.42 – Результати роботи системи в режимі «Веселка»

Висновки до розділу

У ході написання даного розділу були розглянуті бібліотеки мови програмування Rust для взаємодії з послідовними портами: `serial`, `serial-core`, `serial-windows`, `serial-unix`. А також бібліотека для захоплення зображення екрану: `scrap`.

Розглянуто реалізацію драйвера мікроконтролера на базі Arduino Nano. Описано реалізацію програмного забезпечення для операційної системи: програма `AmbiBox` та власне програмне забезпечення.

Було наведено керівництво користувача: налаштування мікроконтролера, програми `AmbiBox` та власного програмного забезпечення.

Показано роботу підсвітки на прикладі домашнього монітору.

5 РОЗРОБКА СТАПТАП-ПРОЕКТУ

5.1 Опис ідеї проекту

5.1.1 Зміст ідеї

В наведеній далі таблиці 5.1 описано зміст ідеї стартап-проекту, а також напрямки його застосування і вигоди, які отримає користувач від проекту.

Таблиця 5.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка драйверу системи управління підсвіткою рекламних оголошень	1. Підсвітка рекламних оголошень	Забезпечення гармонійного сприйняття кольорів як у денний, так і в нічний час для яскравості та запам'ятовування зовнішньої реклами.
	2. Підсвітка монітору персонального комп'ютера	Доповнення ореолом інтенсивність зображення на самому екрані і візуально збільшуючи розмір зображення.
	3. Елемент інтер'єру	Збільшення кількості світла та розширення кольорової гами.

Ідея проекту полягає в розробці драйверу системи управління підсвіткою рекламних оголошень для підвищення якості реклами, покращення загального вигляду рекламних оголошень і збільшення кількості зацікавлених осіб. Драйвер системи також даватиме змогу користувачу гармонійно сприймати кольори в денний, та у нічний час, а також відчувати більшу кількість світла.

5.1.2 Аналіз техніко-економічних переваг ідеї

Система адаптивної підсвітки має наступних конкурентів:

- розробки компанії «Азбука світла»;

- проект управління декоративною підсвіткою на Arduino з телефону;
- Helvar.com – компанія, яка пропонує ідею штучного інтелекту для освітлення.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економічні характерис- тики ідеї	Товари/концепції (потенційні) конкурентів				W (слабка сторона)	N (нейтраль на сторона)	S (сильна сторона)
		Мій проект	Конку- рент1	Конку- рент2	Конку- рент3			
1.	Швидкість роботи	Висока	Висока	Висока	Висока			+
2.	Безвідмов- ність	Висока	Висока	Достатня	Висока		+	
3.	Безпека користувача	Висока	Висока	Висока	Висока		+	
4.	Можливість робота в декількох потоках	+	-	+	-			+
5.	Використан- ня сучасних бібліотек	+	+	+	+		+	
6.	Технології розробки	Новітні	Заста- рілі	Новітні	Новітні		+	
7.	Вартість впроваджен- ня	Низька	Серед- ня	Низька	Висока			
8.	Кросплат- форменність	+	-	-	-			+

Із наведеної вище таблиці можна зробити висновок, що система адаптивної підсвітки має значні переваги над існуючими конкурентами.

5.2 Технологічний аудит ідеї проекту

У таблиці 5.3 наведено технологічну здійсненність ідеї проекту, показано, які технології використовуються для розробки, а також вказано, чи існують такі технології та чи доступні вони.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Розробка драйверу системи управління підсвіткою рекламних оголошень	Rust – мультипарадигмальна компільована мова програмування	Так	Так
		Arduino (мова C) – функціональна мова програмування	Так	Так
		Arduino IDE – інтегроване середовище розробки для Windows, MacOS та Linux, призначена для створення і завантаження програм на Arduino-сумісні плати	Так	Так
		Бібліотеки мови Rust (serial – бібліотека для роботи з послідовними портами, scrap – бібліотека для захоплення екрану)	Так	Так

Таблиця 5.3 дає можливість зробити висновок про те, що даний проект може бути реалізований за допомогою наступних технологій: Rust (особливо, бібліотек – serial та scrap), Arduino (C), Arduino IDE, адже вони наявні і доступні на ринку технологій.

5.3 Аналіз ринкових можливостей запуску стартап-проекту

В таблиці 5.4 наведено аналіз попиту: наявність попиту, обсяг, а також динаміка розвитку ринку.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	2000 грн./ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Перевірка вимог відповідно до постанови Кабінетів міністрів України від 29 грудня 2003 р.
6	Середня норма рентабельності в галузі (або по ринку), %	17

Середню норму рентабельності в галузі порівняно із банківським відсотком на вкладення. Останній являється меншим, тому є сенс вкладати гроші саме у цей проект.

За таблицею 5.4 можна зробити висновок, що ринок є привабливим для входження.

В таблиці 5.5 визначено потенційні групи клієнтів, та їх характеристики.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Базова потреба, яку задовольняє товар (згідно концепції потенційного товару)	Визначити потенційні цільові групи клієнтів, що можуть бути зацікавлені у задоволені означеної потреби	Вписати фактори, що формують поведінку клієнта (стандарты, технічні регламенти, інші фактори цінового і нецінового характеру) та особливості купівлі та експлуатації товару	– до продукції – -до компанії-постачальника
1	Програмне забезпечення для гармонійного сприйняття кольорів як у денний, так і в нічний час	Компанії з розміщення зовнішньої реклами	Відмінності у сферах діяльності компаній та приватних підприємств, підходи до розміщення зовнішньої реклами та створення дизайну інтер'єру	Швидкість та практичність у використанні.
2	Доповнення ореолом інтенсивність зображення на самому екрані та візуально	Активні користувачі персональних комп'ютерів		Можливість кастомізації адаптивної підсвітки

Продовження таблиці 5.5

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	збільшуючи розмір зображення			
3	Збільшення кількості світла та розширення кольорової гами	Компанії з дизайну інтер'єру		Автоматизація процесу підбору кольорової гами

В таблиці 5.6 було проведено аналіз ринкового середовища: наведено фактори загроз, що перешкоджають ринковому впровадженні проекту.

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Вихід на ринок схожих продуктів із кращими характеристиками та реалізованими функціями	Передбачити додаткові переваги власної системи для того, щоб повідомити про них саме після виходу на ринок конкурентів. Вдосконалення технічних функцій власного продукту

Продовження таблиці 5.6

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
2	Зміна потреб користувачів	Користувачам необхідне програмне забезпечення з трохи інакшим функціоналом	Передбачити можливість додавання нового функціоналу до створеної адаптивної системи
3	Незадоволеність користувачів	Незадоволення та незацікавленість користувачів	Вдосконалення популяризації власного продукту

В таблиці 5.7 проведено аналіз ринкового середовища: наведено фактори можливостей, що сприяють ринковому впровадженню проекту.

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Конкуренція	Відсутність аналогічного продукту для українського ринку	Адаптація програмного продукту до українських особливостей
2	Зміна потреб користувачів	З'являються нові методи та бібліотеки, що будуть швидше та ефективніше працювати у системі	Покращити систему додаванням нового функціоналу, розширення можливостей

Продовження таблиці 5.7

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
3	Незадоволеність користувачів	Незадоволення потреб користувачів	Проаналізувати потреби користувачів, а також ринкового середовища

В таблиці 5.8 наведено аналіз пропозиції: визначено ступеневий аналіз конкуренції на ринку.

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія	На ринку присутні декілька компаній-конкурентів, але їх товар дещо відрізняється між собою	Підтримка якості продукту на високому рівні, постійні нововведення та вдосконалення
2. За рівнем конкурентної боротьби - міжнародний	Компанії-конкуренти з інших країн	Створити основу системи таким чином, щоб можна було легко переробити дану для використання її в галузях інших країн
3. За галузевою ознакою - внутрішньогалузева	Продукт може використовуватись для однієї галузі	Постійне удосконалення продукту
4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між видами додатків та їх особливостями	Створити систему адаптивної підсвітки, враховуючи недоліки конкурентів

Продовження таблиці 5.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
5. За характером конкурентних переваг - нецінова	Вдосконалення технології створення системи для того, щоб собівартість була нижчою	Удосконалення готової моделі. Використання ще більш дешевих технологій для розробки, ніж використовують конкуренти
6. За інтенсивністю - марочна	Бренд присутній, і його роль значна	Реклама, участь у конференціях, семінарах

В таблиці 5.9 наведено більш детальний аналіз умов конкуренції в галузі.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
	розробки компанії «Азбука світла», система управління декоративною підсвіткою на Arduino з телефону, Helvar.com	Наявність вже існуючих рішень	-	Різноманітність можливостей	Продукт має більш широкий функціонал, зручніший інтерфейс та автотитет

Продовження таблиці 5.9

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входу в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки:	Інтенсивність конкурентної боротьби з боку прямих конкурентів досить інтенсивна	Можливості виходу на ринок є, але є і конкуренти. Строки – 8 місяців	-	Так, клієнти диктують умови: зручний інтерфейс, швидкість та практичність у використанні.	Потрібно випускати продукт не гірший, ніж у конкурентів

В таблиці 5.10 наведено обґрунтування факторів конкурентоспроможності на основі аналізу конкуренції.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ціна	Більш доступна ціна збільшує кількість потенційних клієнтів

Продовження таблиці 5.10

№ п/п	Фактор конкурентоспроможності і	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
2	Кросплатформність системи	Можливість використання програмного забезпечення на будь-якій платформі.
3	Орієнтованість на кінцевого споживача	Продукт орієнтований на взаємодію з клієнтом

В таблиці 5.11 наведено аналіз слабких і сильних сторін стартап-проекту за визначеними факторами конкурентоспроможності (таблиця 5.10).

Таблиця 5.11 – Порівняльний аналіз сильних і слабких сторін системи адаптивної підсвітки

№ п/ п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні зі стартап-проектом						
			3	2	1	0	1	2	3
1	Ступінь задоволення потреб користувача	15						*	
2	Орієнтація на маркетингову концепцію	7				*			
3	Якість розробленої системи	18							*
4	Ціна	15						*	

В таблиці 5.12 наведено аналіз сильних та слабких сторін, загроз та можливостей стартап-проекту за визначеними ринковими загрозами і можливостями.

Таблиця 5.12 – SWOT-аналіз стартап-проекту

SWOT-аналіз стартап-проекту	
Сильні сторони: Ціна Ступінь задоволення потреб користувача	Слабкі сторони: Якість розробленої системи
Можливості: Поява нових методів та бібліотек, що будуть швидше та ефективніше працювати у системі	Загрози: Зміна потреб користувачів

В таблиці 5.13 наведено альтернативи ринкової поведінки для введення проекту на ринок.

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Комерційне розповсюдження створеної системи	70%	8 місяців
2	Створення системи з подальшим розповсюдженням користувачам за певну оплату за конкретні послуги	35%	8 місяців

5.4 Розроблення ринкової стратегії проекту

На першому кроці розроблення ринкової стратегії передбачено визначення стратегії охоплення ринку. У таблиці 5.14 наведено опис цільових груп потенційних споживачів.

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивніс ть конкуренції в сегменті	Простота входу у сегмент
1	Компанія (українська чи міжнародна) діяльність якої пов'язана з розробкою систем управління адаптивною підсвіткою	Висока Швидкість роботи	40%	Помірна	Готова, так як систему зроблено на високому рівні
2	Будь-яка інша компанія (українська чи міжнародна) діяльність якої пов'язана з розробкою систем управління адаптивною підсвіткою	Висока Яскравий та привабливий дизайн	30%	Помірна	Готова, так як систему зроблено на високому рівні

Продовження таблиці 5.14

№ п/ п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивні сть конкуренц ії в сегменті	Простота входу у сегмент
3	Приватні підприємства міського та міжнародного рівня, діяльність яких пов'язана з розробкою систем управління адаптивною підсвіткою	Помірна Кроссплатформенність	20%	Помірна	Готова, так як систему зроблено на високому рівні
4	Підприємства регіонального характеру	Помірна Логіка роботи системи	10%	Слабка	Готові так як для них це новинка
Які цільові групи обрано: було обрано цільові групи діяльність яких пов'язана з розробкою систем управління адаптивною підсвіткою					

Після вибору цільових груп потенційних споживачів, необхідно сформулювати базову стратегію розвитку. В таблиці 5.15 наведено базову стратегію розвитку.

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспромо жні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Була обрана альтернатива, яка передбачає розробку програмного продукту з подальшим розповсюдженн ям	Визначити потреби кожної з груп, розробити відповідно до них стратегії приваблення клієнтів та маркетингової комунікації	Цінова політика, універсальність продукту (миттєве практичне застосування), орієнтованість на кінцевого споживача	Стратегія диференціації

Далі потрібно обрати стратегії конкурентної поведінки. Базову стратегію конкурентної поведінки наведено в таблиці 5.16.

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохід- цем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	«Прешопрохо- дець»	Компанія шукати нових споживачів	Копіювати основні характеристики товару не буде	Стратегія заняття конкурентної ніші

На основі вимог споживачів із обраних сегментів до стартап-компанії та до продукту (таблиця 5.5), а також залежно від обраної базової стратегії розвитку (таблиця 5.15) та стратегії конкурентної поведінки (таблиця 5.16) розробляється стратегія позиціонування, яка наведена в таблиці 5.17.

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Висока швидкість роботи, кросплатформенність	Стратегія диференціації	Має бути добре захищеною, мати високі вхідні бар'єри; бути непривабливою для конкурентів; відповідати цілям і ресурсам компанії, її специфічним можливостям	Економія часу; зручність застосування; практичність результату

5.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є визначення переваг концепції майбутнього товару. В таблиці 5.18 наведено ключові переваги.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Висока швидкість роботи	Підсвітка швидко змінюється та оновлюється	Відсутність затримок підсвітки при підсвічуванні динамічного зображення.

Продовження таблиці 5.18

№ п/ п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
2	Кросплатформ енність	Програмне забезпечення підходить для будь-якої сучасної операційної системи (Windows та системи на базі Unix)	Відсутність необхідності змінювати операційну систему користувачеві.
3	Практичність результату	Користувач отримує бажаний результат	Користувач на виході роботи отримує задоволення від підсвітки з можливістю кастомізації, а також отримує більшу кількість світла навколо

На другому кроці розробляється та описується трирівнева маркетингова модель товару. В таблиці 5.19 уточнюється ідея продукту і його фізичні складові, а також особливості процесу надання розробленого продукту.

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за здумом	Зручність та швидкість отримання практичного результату щодо процесу виводу підсвітки		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. функція зміни режиму підсвітки	М	Тл
	2. функція адаптації яскравості	М	Тл
	Якість: реалістичність підсвітки		
	Пакування: відсутнє		
	Марка: відсутня		

Продовження таблиці 5.19

Рівні товару	Сутність та складові
III. Товар із підкріпленням	До продажу: відсутнє
	Після продажу: персональна підтримка в обслуговуванні за додаткову плату
За рахунок чого потенційний товар буде захищено від копіювання: за допомогою патенту	

Наступним кроком є визначення меж цін та рівня доходів цільової групи користувачів. В таблиці 5.20 наведено визначення меж встановлення ціни.

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари- замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	0\$	0\$	Середній рівень доходів	0\$

В таблиці 5.21 наведено деталі про систему збуту.

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Цільові клієнти – компанії, котрі бажають впровадити у своїй компанії сучасні засоби, які допоможуть	Встановлення контактів з споживачами та їх підтримка. Формування попиту і стимулювання збуту. Дослідницька робота	Два (від виробник а до телефону, потім до споживач а)	Прямий канал збуту до споживача, потріб но мінімізувати збутові витрати та продовжувати розвиток

Продовження таблиці 5.21

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	отримати кращий дохід. Вони цікавляться сучасними розробками і інноваційними рішеннями.	зі збору маркетингової інформації. Доробка товару, виходячи з потреб конкретного покупця.		маркетингового спілкування зі споживачем

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуютьс я цільові клієнти	Ключові позиції, обрані для позиціонува ння	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Цільові клієнти – компанії, котрі бажають впровадити у своїй компанії	Інтернет-конференції, інноваційні семінари, огляд професійної літератури,	Позиція на основі порівняння фірми із товарами конкурентів	Збільшення потоків покупців та обсягів продажу підсвіток; нормалізація обсягів продажу	Не витрачайте час і гроші на неякісні підсвітки для свої реклами!

Продовження таблиці 5.22

№ п/п	Специфіка поведінки цілових клієнтів	Канали комунікацій, якими користуютьс я цілові клієнти	Ключові позиції, обрані для позиціонува ння	Завдання рекламного повідомлення	Концепція рекламного звернення
1	сучасні засоби, які допоможуть отримати кращий дохід. Вони цікавляться сучасними розробками і інноваційними рішеннями.	інтернет, спілкування з іншими компаніями з розробки систем управління адаптивними підсвітками.		в період зменшення попиту та загального зменшення ділової активності.	

Висновки до розділу

У даному розділі було наведено зміст ідеї стартап-проекту, а також напрямки його застосування та вигоди, які отримає користувач від проекту, проведено аналіз техніко-економічних переваг ідеї. Проведено технологічний аудит ідеї проекту. Також проведено аналіз ринкових можливостей запуску проекту.

Було розроблено ринкову стратегію проекту, а також маркетингову програму стартап-проекту.

ВИСНОВКИ

У ході виконання магістерської дисертації розглянуті питання, які виникають у процесі оформлення зовнішньої реклами. Було виділено важливе практичне завдання – підвищення якості реклами, покращення загального вигляду рекламних оголошень та збільшення кількості зацікавлених осіб.

У першому розділі було наведено основні відомості щодо предметного середовища – зовнішньої реклами. Був також проведений докладний аналіз предметного середовища. Розглянуто основні аналоги системи, виділено їх переваги та недоліки. Також наведено постановку задачі.

Для розробки драйвера мікроконтролера було використано такі засоби для програмування на Arduino як Arduino IDE, а для розробки програмного забезпечення для налаштування і персоналізації фонові підсвітки використано мову програмування Rust.

Для взаємодії з послідовними портами було обрано бібліотеки мови програмування Rust: serial, serial-core, serial-windows, serial-unix. А також бібліотека для захоплення зображення екрану: scrap.

Наведено детальне керівництво користувача по експлуатації системи підсвітки, описано необхідні налаштування, які показують можливість введення системи підсвітки в експлуатацію.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Підсвітка рекламних оголошень. [Електронний ресурс] – Режим доступу до ресурсу: https://svetpro.ru/htm/informations/info_133.html
2. Професійна підсвітка рекламних оголошень. [Електронний ресурс] – Режим доступу до ресурсу: <https://5watt.ua/blog/stati/professionalnaya-podsvetka-billbordov-i-reklamnykh-sshitov>
3. Азбука світла. [Електронний ресурс] – Режим доступу до ресурсу: <https://abclight.ru/infrastructural-lighting/advertisement-lighting/>
4. Керування декоративною підсвіткою з телефону [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/165861/>
5. Інтелектуальні світлові рішення [Електронний ресурс] // Режим доступу: <https://helvar.com/>
6. Послідовний порт [Електронний ресурс] – Режим доступу до ресурсу: <https://dic.academic.ru/dic.nsf/ruwiki/77526>
7. Світлодіодні стрічки [Електронний ресурс] – Режим доступу до ресурсу: <https://arduinomaster.ru/datchiki-arduino/adresnaya-svetodiodnaya-lenta/>
8. Мова програмування Ардуіно: основні поняття [Електронний ресурс] – Режим доступу до ресурсу: <https://arduinoplus.ru/arduino-yazyk-programmirovaniya/>
9. Мови C і C++. Де їх використовують і навіщо? [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/nuances-of-programming/языки-с-и-с-где-ещё-их-используют-и-зачем-9ff56559d5bf>
10. AVR Libc [Електронний ресурс] – Режим доступу до ресурсу: <https://curlie.org/Computers/Hardware/Components/Processors/AVR/>
11. Програмування Arduino [Електронний ресурс] – Режим доступу до ресурсу: <https://all-arduino.ru/programmirovanie-arduino/>
12. Документація Arduino [Електронний ресурс] – Режим доступу до ресурсу: <https://support.arduino.cc/hc/en-us#toc13>
13. Arduino Nano: схема підключення та програмування [Електронний ресурс] –

Режим
доступу
до
ресурсу:

<http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:arduino-nano>

14. Мова програмування Rust [Електронний ресурс] – Режим доступу до ресурсу: https://geekbrains.ru/posts/why_rust

15. Бібліотека serial [Електронний ресурс] – Режим доступу до ресурсу: <https://crates.io/crates/serial>

16. Бібліотека serial-core [Електронний ресурс] – Режим доступу до ресурсу: <https://crates.io/crates/serial-core>

17. Бібліотека serial-windows [Електронний ресурс] – Режим доступу до ресурсу: <https://crates.io/crates/serial-windows>

18. Бібліотека serial-unix [Електронний ресурс] – Режим доступу до ресурсу: <https://crates.io/crates/serial-unix>

19. Бібліотека scrap [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/quadrupleslap/scrap>

20. Бібліотека анімації FastLED [Електронний ресурс] – Режим доступу до ресурсу: https://alexgyver.ru/fastled_vs_neopixel/

21. AmbiBox [Електронний ресурс] – Режим доступу до ресурсу: http://www.ambibox.ru/ru/index.php/Заглавная_страница